UNITED STATES DISTRICT COURT
EASTERN DISTRICT OF NEW YORK

ARTEC EUROPE S.A.R.L.,

Plaintiff,

v.

SHENZHEN CREALITY 3D TECHNOLOGY CO., LTD., AND KICKSTARTER, PBC,

Defendants.

1:22-1676 (WFK)(VMS)

DECLARATION OF GEORGE EDWARDS, PH.D.

1. INTRODUCTION

1. My name is George Edwards. I have been retained as an expert in computer science and source code analysis to provide my opinions in connection with the matter of *Artec Europe S.a.r.l. v. Shenzhen Creality 3D Technology Co., Ltd. And Kickstarter, PBC*. I was asked by counsel for Artec Europe S.a.r.l. ("Artec") to provide my expert opinion on:

- The extent to which a software application developed by Artec named Artec Studio is similar to two other software applications, named Magic Wand and CR Studio.
- Whether certain evidence indicates that the same source code was used in components of Artec Studio, Magic Wand, and CR Studio.
- What, if any, conclusions can be drawn regarding the likelihood that source code was copied from Artec Studio into Magic Wand and CR Studio.

2. In forming my opinions, I analyzed the functionality, user interfaces, and software files of Artec Studio, Magic Wand, and CR Studio. I submit this declaration as a statement of the opinions that I have formed and the factual basis for those opinions. If necessary, I am prepared to testify as to the matters discussed in this report.

3. I am being compensated for my work on this case at the rate of $575 per hour plus reimbursement of direct expenses. I was assisted in portions of my analysis by my colleague, Dr. Hassan Afzali-Kusha. I have no personal interest in this litigation, and my compensation does not depend in any way on the opinions I express or outcome of this case.

2. QUALIFICATIONS

4. This section summarizes my education, professional achievements, and peer-reviewed scientific publications in the field of computer science. A more detailed list of my qualifications is set forth in my curriculum vitae, a copy of which is attached hereto as **Exhibit A**.

2.1. Education

5. I graduated *summa cum laude* with a Bachelor of Science degree in computer science from Vanderbilt University in 2003. As an undergraduate, I was awarded the Vanderbilt School of Engineering Merit Scholarship. I spent one year in the MS program in computer science at Vanderbilt. During this time, I conducted research on software modeling and visualization, and I published my findings in several peer-reviewed conference papers and journal articles.

6. I attended graduate school at the University of Southern California, where I was a USC Viterbi School of Engineering Dean's Doctoral Fellow and Annenberg Graduate Fellow. I received an MS in computer science in 2006 and a PhD in computer science in 2010 from USC. My PhD research focused on software design and analysis. My research was funded by several government agencies, including the Department of Defense and the NSA, and large companies such as Bosch

and InfoSys. I presented my work at numerous conferences and in academic journals, industry magazines, and other publications. In 2008, I received the USC Computer Science Department's award for outstanding graduate student research.

2.2. Employment

7. I currently am employed as Principal Computer Scientist at Quandary Peak Research, Inc. in Los Angeles, CA. In this role, I analyze the development, structure, behavior, and quality of software systems. I have analyzed a broad variety of complex, real-world software systems, and I have conducted many investigations of the design and implementation of these systems in the context of intellectual property litigation. I have analyzed many software systems and their code to determine whether every element of one or more patent claims is found in the system. I also have analyzed software to determine whether it incorporates designs, functions, algorithms, data structures, or other internal structures or behaviors that are alleged to be trade secrets. In copyright infringement litigation, I have conducted numerous comparisons of one application to another to determine the nature, degree, and scope of similarity between them.

8. I formerly was employed as a Lecturer of Computer Science at the University of Southern California. In that capacity, I taught Requirements Engineering (CSCI 568), a graduate-level software engineering class. I also taught Data Structures and Algorithms (CSCI 102), an undergraduate-level software design and programming class.

9. I also formerly worked as a research scientist and software engineer at Blue Cell Software LLC, Intelligent Systems Technology, Inc., IBM, and The Boeing Company. During my time with Blue Cell, I built a simulation-based software design and modeling environment. While at IBM, I conducted research on next-generation mobile architectures, such as large-scale mobile device

provisioning systems. At Boeing, I helped to design a specialized mobile computing device for a robust and survivable peer-to-peer wireless network.

2.3. Publications

10. I have authored over thirty scholarly journal articles, magazine articles, conference papers, and book chapters on varied topics related to software engineering and distributed systems. I have delivered numerous invited lectures, seminars, and technology demonstrations related to software design and analysis for university courses, research symposia, conferences, workshops, and industry events in the field of computer science. I also have served as a reviewer, committee member, or panelist for over a dozen computer science journals, magazines, and conferences.

2.4. Prior Testimony

11. On July 17, 2013, I testified by deposition in the matter of *Essociate, Inc. v. Neverblue Media, Inc.*, a patent case in the U.S. District Court for the Central District of California.

12. On November 25, 2013, I testified in U.S. District Court for the Eastern District of Virginia as an expert in mobile devices in a patent infringement matter, *Porto Technology Co., Ltd. and Porto Technology, LLC v. Cellco Partnership d/b/a Verizon Wireless and Verizon Services Corp.*

13. On June 9, 2014, I testified by deposition in the matter of *SecurityBase.com v. Jeffrey Essick, et al.*, a breach-of-contract suit brought in the Superior Court of California for the County of Orange.

14. I testified both by deposition (on June 17, 2014) and in U.S. District Court for the District of Kansas (on June 24, 2014) in the matter of *AgJunction, LLC v. Agrian, Inc. et al.*, a copyright and trade secret case.

15. I testified both by deposition (on October 12, 2015) and in U.S. District Court for the Southern District of California (on December 7 and 8, 2015) in the matter of *Anthony Johnson v. Storix, Inc.*, a copyright infringement matter.

16. On August 15, 2016, I testified by deposition in the matter of *Backflip Software, Inc. v. Cisco Systems, Inc. et al.*, a breach-of-contract suit brought in the Superior Court of California for the County of Santa Clara.

17. On July 12, 2017, I testified by deposition in the matter of *Implicit, LLC v. Trend Micro, Inc.*, a patent case in the U.S. District Court for the Eastern District of Texas.

18. On December 8, 2017, I testified by deposition in the matter of *Farmers Edge Inc., Farmers Edge (US) Inc., and Farmers Edge (US) LLC v. Farmobile, LLC, Jason G. Tatge, Heath Garrett Gerlock, and Randall Thomas Nuss*, a trade secret misappropriation case in the U.S. District Court for the District of Nebraska.

19. On December 14, 2017, I testified by deposition in the matter of *Vesta Corporation v. Amdocs Management Limited, et al.*, a trade secret misappropriation case in the U.S. District Court for the District of Oregon.

20. On December 18, 2017, I testified by deposition in the matter of *eBay, Inc. v. MasterObjects, Inc.*, an *inter partes review* proceeding before the Patent Trial and Appeal Board of the United States Patent and Trademark Office.

21. On May 23, 2018, I testified in U.S. District Court for the District of Nebraska in the matter of *Farmers Edge Inc., Farmers Edge (US) Inc., and Farmers Edge (US) LLC v. Farmobile, LLC, Jason G. Tatge, Heath Garrett Gerlock, and Randall Thomas Nuss*.

22. On June 28, 2018, and May 25, 2021, I testified by deposition in the matter of *Bryndon Fisher v. The United States of America*, a class action lawsuit in the U.S. Court of Federal Claims.

23. On December 5-6, 2019, I testified by deposition in the matter of *American National Manufacturing, Inc. v. Sleep Number Corporation*, an *inter partes review* proceeding before the Patent Trial and Appeal Board of the United States Patent and Trademark Office.

3. SUMMARY OF OPINIONS

24. Based on my analysis of the materials provided to me in this matter, it is my opinion that:

- Magic Wand and CR Studio exhibit an idiosyncratic, difficult-to-detect, and undesirable behavior (*i.e.*, misbehavior) found in Artec Studio. This misbehavior is almost certainly caused by the same bug in the source code of all three applications. For the reasons explained in this report, the most plausible and apparent explanation for the existence of the bug in Magic Wand and CR Studio is that source code including the bug was copied from Artec Studio into Magic Wand and CR Studio.
- It is likely that the Magic Wand and CR Studio software contain decoding algorithm code that is a copy of, was derived from, or was written with reference to the Artec Studio decoding algorithm code for the Artec Eva scanner. The decoding algorithm is fundamental to the Artec Studio software's function and purpose.
- I confirmed that there are similarities in portions of the user interface as described in the Gusev Declaration. In my opinion, the similarity in the Workspace/Data Panel windows of Artec Studio, Magic Wand, and CR Studio would be unlikely to occur purely by chance.

4. MATERIALS CONSIDERED

25. In the course of forming my opinions, I reviewed and considered the following materials:

- Complaint of Plaintiff Artec Europe S.a.r.l.

- Memorandum of Law in Support of Plaintiff Artec Europe S.a.r.l.'s Application for a Temporary Restraining Order and Preliminary Injunction
- U.S. Patent 7,768,656 ("the '656 Patent")
- Declaration of Gleb Gusev dated March 25, 2022 ("Gusev Declaration")
- Videos linked from the Gusev Declaration
- Shenzhen Creality 3D Technology Co., Ltd's Memorandum of Law in Opposition to Plaintiff's Application for a Temporary Restraining Order and Preliminary Injunction
- Declaration of Xun Luo ("Luo Declaration")
- Declaration of Luowei ("Luowei Declaration")
- Kickstarter, PBC's Memorandum of Law in Opposition to Plaintiff's Application for a Temporary Restraining Order and Preliminary Injunction
- Executable installation file for Artec Studio version 9.2.2.34-x64
- Executable installation file for Magic Wand version 1.3.6.009280
- Executable installation file for CR Studio version 1.7.4.383
- Calibration and object files for the Artec Studio, Magic Wand, and CR Studio applications
- Declaration of Gleb Gusev dated April 6, 2022 ("Supplemental Gusev Declaration")
- Scientific literature, authoritative texts, and other references in the field of the claimed technologies, where cited in this report

5. ANALYSIS METHODS

26. Over the course of my career as a computer scientist specializing in software analysis, I have conducted numerous comparisons of software applications and programs in the context of copyright litigation. I have performed comparisons to determine the nature, degree, and scope of

similarity between applications. For example, I have conducted comparisons of the relevant applications' structure, sequence, and organization in terms of their source code, object code, database schemas, user interfaces, component structure, technology stack, and other design decisions. I understand the abstraction-filtration-comparison (AFC) test and have performed the AFC test. I have applied different comparison methods and tools depending on the materials available for analysis, the nature of the alleged copying, the type of software to be compared, and other factors. This section explains the analysis methods and tools I employed in the course of forming the opinions expressed in this report.

27. To perform my analysis in this matter, I was provided with the installation files for Artec Studio 9, Magic Wand 1.3.6, and CR Studio 1.7.4. I compared the user interfaces, functions, and installed files of these applications. As explained more fully in Section 6 below, my analysis focused on distinctive, idiosyncratic features and behaviors in the software.

28. I have not yet been provided with any source code for Artec Studio, Magic Wand, or CR Studio, so I have not yet been able to perform a source code comparison. I believe a rigorous source code comparison would provide additional information relevant to the opinions I give in this report. For example, I believe a source code comparison would clarify the nature, volume, and scope of source code that has been copied from Artec into Magic Wand and CR Studio. I therefore reserve my right to amend or supplement this report based on source code or other additional material that may be provided to me in the future.

5.1. User Interface Comparison

29. I compared the user interfaces of Artec Studio, Magic Wand, and CR Studio by installing and running the products. I captured screenshots of the user interfaces of each product. I compared the user interfaces by examining the structure, sequence, and organization of the user interface

elements. The user interface elements, which commonly are referred to as "widgets," include windows, lists, buttons, labels, and so on. My examination focused on the combination of widgets that are present in each screen and how those widgets are arranged.

5.2. Functional Comparison

30. I installed and executed Artec Studio, Magic Wand, and CR Studio to compare the functions and features of Artec Studio with the functions and features of Magic Wand and CR Studio. This type of software analysis is known as "dynamic analysis." Dynamic analysis includes both automated and manual techniques for studying a computer program by executing the program to monitor its behavior and capture its responses to various inputs. By executing Artec Studio, Magic Wand, and CR Studio, I was able to view the various options, menus, and features available in each product, and see the results of selecting those options, menus, and features as a way to compare the functions and features of the products.

5.3. Examination of Application Files

31. I examined the application files created by Artec Studio, Magic Wand, and CR Studio during installation on a computer running Microsoft Windows. The application files included binary executables and libraries, as well as numerous other auxiliary files such as configuration files, user documentation, and image files, among others.

6. ANALYSIS RESULTS

32. My analysis determined that several distinctive features and behaviors of Artec Studio were also present in Magic Wand and CR Studio.

6.1. Anomalous "Click-and-Drag" Behavior

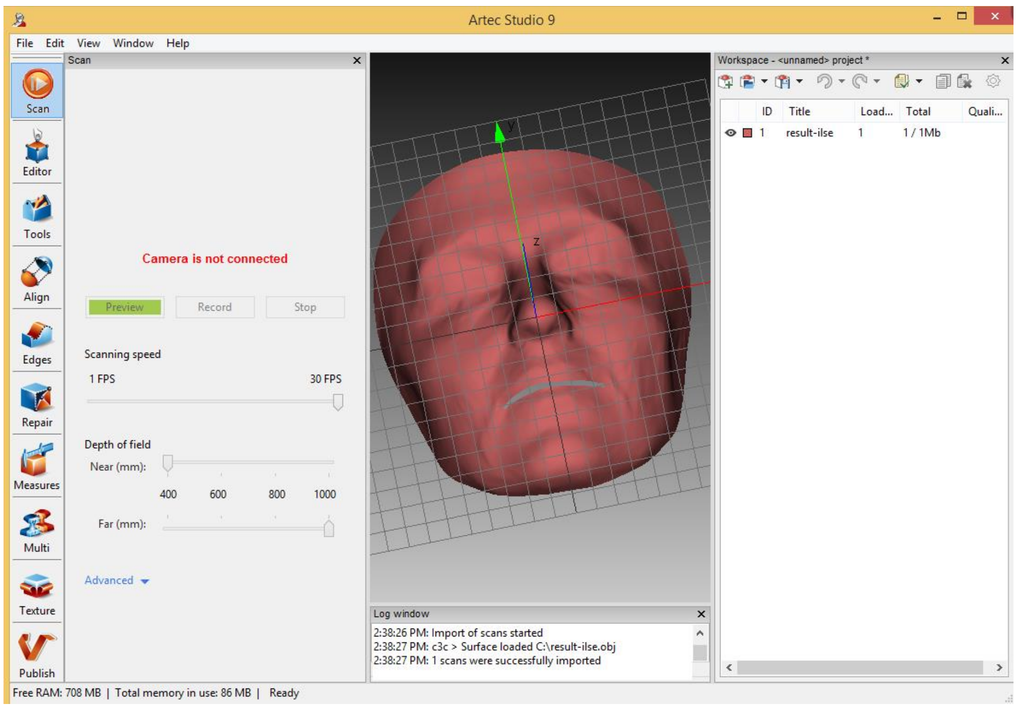
33. I executed and used Artec Studio, Magic Wand, and CR Studio in order to observe the functionality and behavior of each product. I personally reproduced and confirmed the anomalous

behavior described in the Gusev Declaration at paragraph 16. As described in the Gusev Declaration, the anomalous behavior occurs "when a user 'drags' an object by performing a circular motion with the mouse." When this action is performed in Artec Studio, Magic Wand, and CR Studio, the normal behavior intended by the developers and expected by users does not occur. As I elaborate more fully below, the presence in all three applications of the same *mis*behavior is highly significant precisely because it is peculiar, difficult-to-detect, and undesirable. The most plausible explanation for the observed misbehavior is that at least some source code was copied from Artec Studio into Magic Wand and CR Studio.

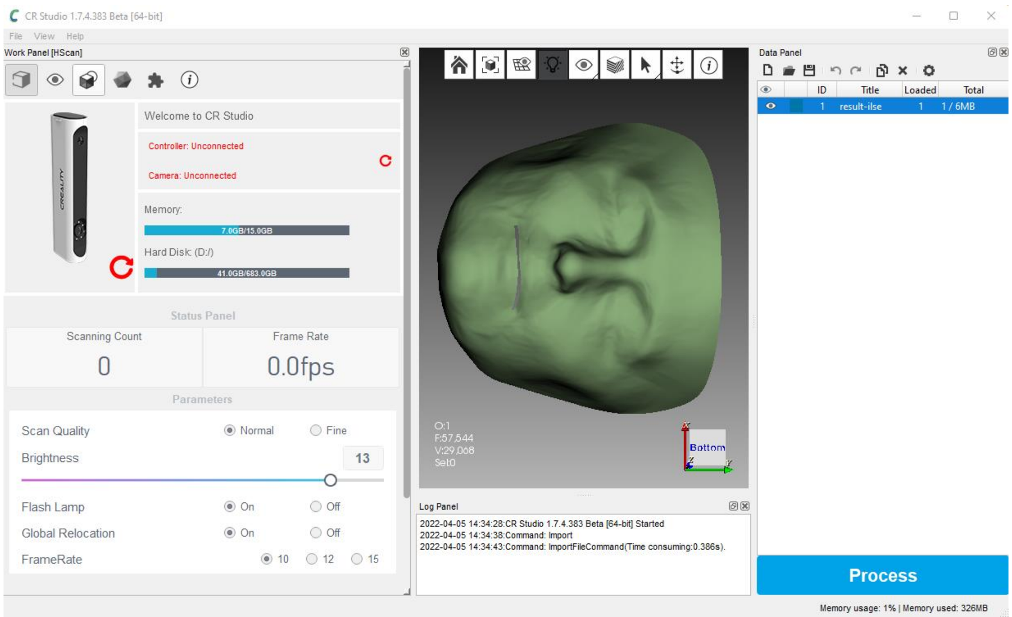
34. Specifically, Artec Studio, Magic Wand, and CR Studio include the capability for a user to rotate the 3D model of a scanned object using the familiar and intuitive click-and-drag behavior. Screenshots of this capability are shown below. Under normal usage scenarios, all three applications perform the click-and-drag action in the intended and expected way that other applications typically do.

35. The intended and expected behavior is as follows: when a user drags an object such that the dragging movement ends with the mouse cursor back in the original position where it started, the object being dragged should also return to its original position on the screen. This is the intended behavior with which most computer users are familiar. For example, in the Microsoft Windows operating system, a user may drag an application window (*e.g.*, a web browser window) by clicking-and-dragging on the window's title bar. If the user dragged the window to the left and then back to the right, ending with the mouse cursor in the original position where it started, the dragged window would move with the mouse to the left and then back to the right, and the window would eventually position back in its original location.

36. Under a specific set of conditions, however, Artec Studio, Magic Wand, and CR Studio all exhibit an identical behavior that is not intended or expected with regard to the rotation of a 3D model. Just like any other program behavior, the anomalous behavior is caused by the program's source code. In this instance, the unintended and unexpected behavior is caused by a defect, or "bug," in the program source code.[1]



[1] Gusev Declaration, ¶16, characterizing the behavior as "counterintuitive and dysfunctional."



37. The anomalous behavior caused by the source code bug in Artec Studio, Magic Wand, and CR Studio occurs as follows: when a user drags an object in a circular path, returning to the origin of the dragging motion, the object rotates nearly, but not quite, back to the original orientation. Rather than returning exactly back to its original orientation as it should, the object's orientation has essentially "drifted" slightly during the click-and-drag action. If the circular click-and-drag action is performed only once, the "drift" is small, and it is likely most users would not notice it. However, if the circular click-and-drag action is performed repeatedly, the "drift" accumulates each time, and after many such repeated actions, the object's orientation will have changed noticeably.

38. Using Artec Studio, Magic Wand, and CR Studio, I personally confirmed that this behavior – and by extension the bug that causes it – is present in all three applications and causes all three of them to misbehave in the same way. I now will turn to a discussion of why this finding is important.

39. During an examination of software for evidence of copying, it is common practice for software analysts to pay particular attention to the presence of idiosyncratic mistakes, typos, errors,

bugs, and other defects because it is unlikely such bugs or defects would appear independently in two different programs by coincidence.[2] Since bugs and defects are undesirable, a programmer would not purposefully include them (*i.e.*, "the behavior could not be motivated by the task"[3]). Moreover, since human programmers can make minor errors in countless different ways, there are innumerable different bugs that might crop up in any given software program. For the same rare bug to appear in two separate programs, there may be only one plausible explanation: the code of one program was copied into the other. Of course, while the presence of common errors in both programs is very strong evidence that some copying has occurred, a complete source code analysis still is necessary to determine the nature, volume, and scope of source code that has been copied.

40. Based on my examination of Artec Studio, Magic Wand, and CR Studio, and my observation of the bug in question, it is my opinion that it would be virtually impossible for this particular bug to occur in all three applications by coincidence. Based on the information currently available to me, I believe the only reasonable explanation for this specific misbehavior appearing in all three applications is that all three applications are using the same source code containing the very same bug (or, alternatively, code that is closely derived from that code) to perform the click-and-drag 3D object rotation.

41. There are, therefore, three reasonably plausible possibilities:

(1) the code containing the bug was copied from Artec Studio into Magic Wand/CR Studio;

(2) the code containing the bug was copied from Magic Wand/CR Studio into Artec Studio; or

---

[2] *See*, *e.g.*, Robert Zeidman, *The Software IP Detective's Handbook*, Prentice Hall (2011), p. 330.

[3] Amicus Brief of Computer Scientists, Harbor Software v. Applied Systems, No. 97-7197 (2d Cir. 1998), p. 16: "In many instances, the initial evidence of copying is a relatively insignificant incident that turns out to be only the tip of the iceberg. For example, one case that eventually involved claims of extensive code theft (both literal and non-literal) as well as a variety of other serious charges began when a software developer noticed that a competing program built by former employees of his company displayed the same misbehavior that he knew to be present in his own code. As a piece of misbehavior, the behavior could not be motivated by the task, suggesting the possibility of copying." (**Exhibit B**)

(3) the code containing the bug was copied from some other external source into all three applications.

42. Possibility (2) can be ruled out because Artec Studio predates both Magic Wand and CR Studio.[4] For example, Artec Studio version 9.2 was released in 2013,[5] whereas Jimuyida was not even founded until 2015.[6] Also, it is my understanding that there is currently no known or alleged channel through which Artec could have accessed the code to Magic Wand or CR Studio.[7]

43. Possibility (3) may be ruled out because the Supplemental Gusev Declaration states that the code that implements the click-and-drag misbehavior is proprietary, confidential code developed by Artec, and the code does not come from any third-party library or other external source.[8] Such proprietary, confidential code is typical of closed-source, commercial applications like Artec Studio.

44. Thus, the most plausible and apparent explanation for the observed misbehavior is that at least some source code was copied from Artec Studio into Magic Wand and CR Studio. Given the presence of other similarities discussed in greater detail below, it is my conclusion that more code than just the 3D object rotation code was likely copied.

6.2. Structured Light Pattern

45. I inspected the application files created by the installation of Artec Studio, Magic Wand, and CR Studio on a Windows computer. During this inspection, I identified image files within the application folder of CR Studio that depict a distinctive structured light pattern. Also, the

[4] Gusev Declaration, ¶7.
[5] See https://www.artec3d.com/news/artec-studio-92-out-introducing-auto-align-tool-and-much-more (**Exhibit C**)
[6] Luowei Declaration, ¶3.
[7] Shenzhen Creality 3D Technology Co., Ltd's Memorandum of Law in Opposition to Plaintiff's Application for a Temporary Restraining Order and Preliminary Injunction; Luo Declaration; Luowei Declaration; Kickstarter, PBC's Memorandum of Law in Opposition to Plaintiff's Application for a Temporary Restraining Order and Preliminary Injunction.
[8] Supplemental Gusev Declaration, ¶3.

Supplemental Gusev Declaration states that the structured light pattern projected by the Magic Wand scanner is nearly identical to that projected by the Artec Eva scanner.[9] The projected structured light pattern is a function of the slide pattern included in the scanner. As I explain in this section, the use of substantially similar slide patterns in the Artec Eva scanner and Magic Wand scanner means that the Magic Wand and CR Studio applications contain code that implements a decoding algorithm substantially similar to the one in Artec Studio. This represents a significant functional similarity between Artec Studio and Magic Wand/CR Studio because the decoding algorithm is fundamental to the software's purpose and is necessary for the 3D scanner to function.

46. A 3D object scanner projects a structured light pattern onto the surface of an object.[10] More specifically, a light projector in the 3D object scanner may include a slide pattern to project the structured light pattern, as depicted in Fig. 3 of the '656 Patent. For the 3D object scanner to function correctly and efficiently, the structured light pattern must be designed carefully in concert with a decoding algorithm implemented in software and other aspects of the system's design.[11] The code that implements the decoding algorithm is strongly dependent on the specific slide pattern of the scanner.

47. At the same time, there are many structured light patterns that may be used in a 3D object scanner. As long as the structured light pattern has the required characteristics, the corresponding decoding algorithm code can be written to function correctly.[12] In other words, while the slide pattern used in the Artec Eva is not the only pattern that works, it is the only pattern that works

---

[9] Supplemental Gusev Declaration, ¶8.
[10] Webster, J.G., Bell, T., Li, B. and Zhang, S. (2016). Structured Light Techniques and Applications. In Wiley Encyclopedia of Electrical and Electronics Engineering, J.G. Webster (Ed.). (**Exhibit D**)
[11] '656 Patent.
[12] '656 Patent, 6:35-62.

with Artec's Eva decoding algorithm code. If Creality desired to use a copy or near copy of Artec's decoding algorithm code in Magic Wand and CR Studio, Creality would necessarily have to use a slide pattern that was the same as or substantially similar to the slide pattern of an Artec scanner. If, on the other hand, Creality desired to design both a slide pattern and accompanying decoding algorithm from scratch, it would be (1) unnecessary for them to copy the Artec Eva slide pattern, and (2) unlikely that they would independently design a slide pattern that was nearly identical to that of the Artec Eva.

48. I identified image files within the application folder of CR Studio that depict a distinctive structured light pattern. These image files were found in subfolders of a folder named "benchMarkData." One of the image files found in the CR Studio "benchMarkData" application folder is shown below. Although I did not find these image files in the application folder of Magic Wand, there is a high degree of similarity between the application files of Magic Wand and CR Studio generally.[13]

[13] Gusev Declaration, ¶28.

49. Based on these observations and findings, it is my opinion that it is likely that the Magic Wand and CR Studio software contain decoding algorithm code that is a copy of, was derived from, or was written with reference to the Artec Studio decoding algorithm code for the Artec Eva scanner. As noted earlier, the decoding algorithm is fundamental to the Artec Studio software's purpose and is necessary for the 3D scanner to function.

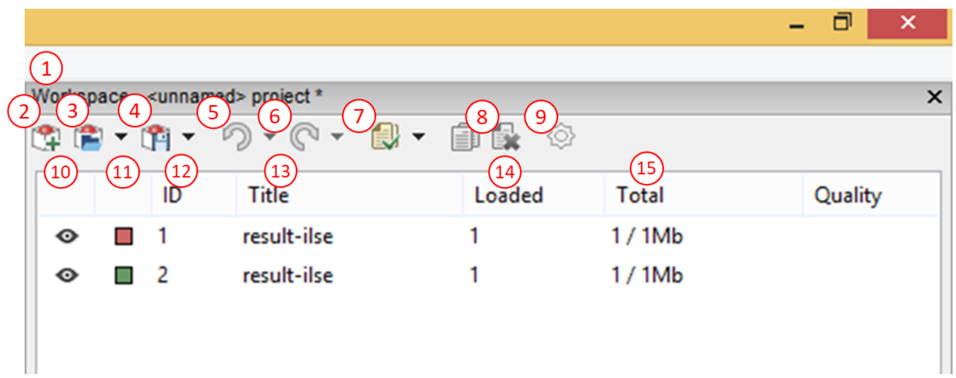
6.3. User Interface Comparison

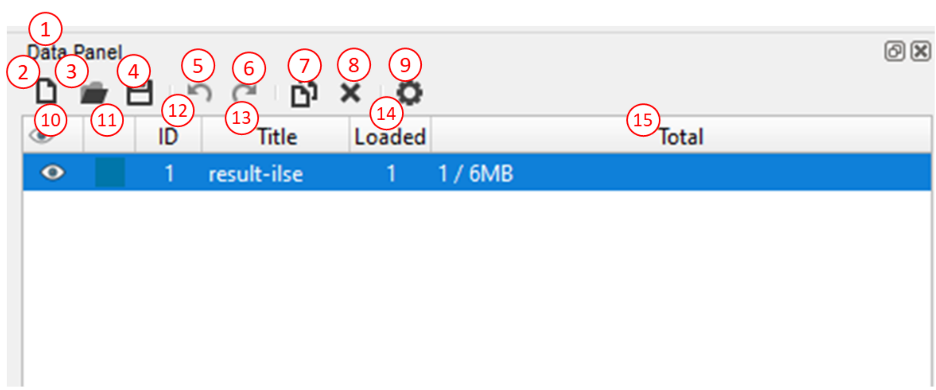
50. I compared the user interfaces of Artec Studio, Magic Wand, and CR Studio. I confirmed that there are similarities in portions of the user interface as described in the Gusev Declaration.[14]

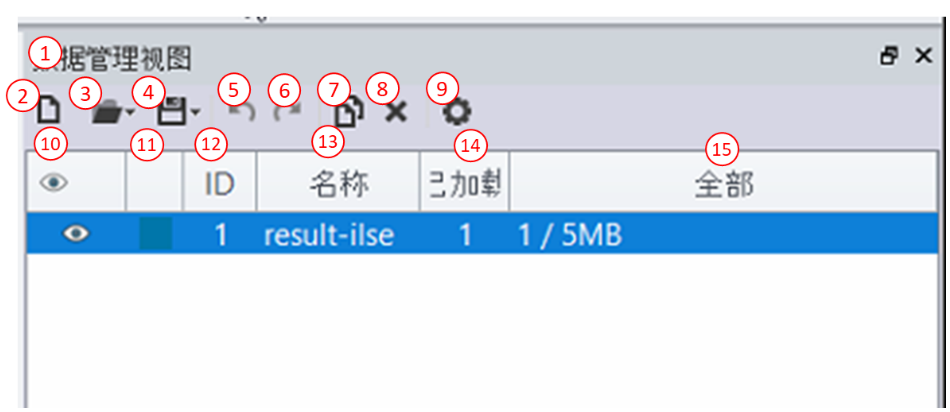
[14] Gusev Declaration, ¶22-23.

For example, I observed that all three applications include a "Workspace" or "Data Panel" window that is docked to the right side of the main application window by default. In all three applications, the Workspace/Data Panel window includes a toolbar at the top containing a nearly identical sequence of buttons. The main window is occupied by a table containing a nearly identical sequence of columns. The rows in the table correspond to objects. Screenshots of the Workspace/Data Panel windows of Artec Studio, CR Studio, and Magic Wand are shown below, with numbered labels added to indicate the similar sequence of common elements. In my opinion, the similarity in the Workspace/Data Panel windows of Artec Studio, Magic Wand, and CR Studio would be unlikely to occur purely by chance.

1. Workspace/Data Panel
2. New Project
3. Open Project
4. Save Project
5. Undo
6. Redo
7. Select Scans
8. Delete Scans
9. Settings
10. Show/Hide
11. Color
12. ID
13. Title
14. Loaded
15. Total







7. CONCLUSIONS

52. Based on my analysis, I reached the following conclusions:

- Magic Wand and CR Studio exhibit an idiosyncratic, difficult-to-detect, and undesirable behavior (*i.e.*, misbehavior) found in Artec Studio. This misbehavior is almost certainly caused by the same bug in the source code of all three applications. For the reasons explained in this report, the most plausible and apparent explanation for the existence of the bug in Magic Wand and CR Studio is that source code including the bug was copied from Artec Studio into Magic Wand and CR Studio.
- It is likely that the Magic Wand and CR Studio software contain decoding algorithm code that is a copy of, was derived from, or was written with reference to the Artec Studio decoding algorithm code for the Artec Eva scanner. The decoding algorithm is fundamental to the Artec Studio software's function and purpose.
- I confirmed that there are similarities in portions of the user interface as described in the Gusev Declaration. In my opinion, the similarity in the Workspace/Data Panel windows of Artec Studio, Magic Wand, and CR Studio would be unlikely to occur purely by chance.

I declare under penalty of perjury that the foregoing is true and correct.

Executed on 4/6/2022 at Los Angeles, California.

____

George Edwards

EXHIBIT A

GEORGE EDWARDS

Computer Scientist
Quandary Peak Research
205 S Broadway, Ste 300
Los Angeles, CA 90012
Phone: 323.545.3933
Email: george@quandarypeak.com

EDUCATION

- **Ph.D. in Computer Science**, *University of Southern California* **Aug 2010**
- **M.S. in Computer Science**, *University of Southern California* **May 2006**
- **B.S. in Computer Science** with Minor in Mathematics, *Vanderbilt University* **Aug 2003**

EMPLOYMENT

- **President & Principal Computer Scientist** **June 2012 – Present**
  Quandary Peak Research, Los Angeles, CA
  - Providing software analysis and expert witness testimony in software-related litigation, including patent and copyright infringement, theft of trade secrets, breach-of-contract, and other matters.
  - Analyzing software intellectual property and patent portfolios for validity and infringement in the context of licensing and brokering negotiations, startup investments, and M&A.
  - Investigating software failures to determine the root cause and help clients understand whether and how the failure could have been avoided.
  - Applying advanced analytic techniques based on calibrated parametric models for valuation of software products and estimation of software development costs.
  - Documenting the architecture of software systems to identify structural similarities and differences among competing products and services and deduce the origin of software designs and code.

- **Lecturer of Computer Science** **Jan 2012 – May 2017**
  Computer Science Department, University of Southern California, Los Angeles, CA
  - Teaching a graduate-level software engineering course for M.S. and Ph.D. students and practicing software and aerospace engineers (CSCI 568).
  - Previously taught an undergraduate-level course on programming fundamentals, algorithms, and data structures (CSCI 102).

- **Co-Founder and CEO** **Aug 2010 – June 2012**
  Blue Cell Software, Los Angeles, CA
  - Led and managed the development of a next-generation modeling and analysis platform for complex software-intensive systems.
  - Developed advanced model-based algorithms for reliability, efficiency, and risk analysis.
  - Conceived and implemented the business development strategy and produced and presented investment proposals.

- **Post-Doctoral Researcher** **Feb 2011 – Sept 2011**
  Center for Systems and Software Engineering, University of Southern California, Los Angeles, CA
  - Conducted research on enterprise integration mechanisms to support net-centric operations.
  - Investigated methods, processes, and tools for decomposition and refinement of SoS capability needs into detailed requirements and integration architectures for an NSA-funded research project.
  - Designed key components of PATFrame, a decision support system for planning test events of networks of unmanned and autonomous vehicles.
  - Applied lessons learned from large-scale corporate mergers, including the HP-Compaq merger, to software and systems integration challenges faced by the DoD and NSA.

- **Graduate Fellow** **Aug 2004 – Aug 2010**
  Computer Science Department, University of Southern California, Los Angeles, CA
  - Designed and implemented XTEAM, a model-driven software design, analysis, and code generation toolset. XTEAM allows software architects to create domain-specific software architecture models, evaluate those models with respect to quality metrics, and automatically synthesize custom code. XTEAM has been utilized by more than a dozen software engineering projects in academia and industry.
  - Built key components of Prism-MW, a novel Java-based implementation platform for mobile and handheld devices that enables rapid development of event-based applications.
  - Invented the adaptive layered style, an architectural style for autonomous systems, and designed the corresponding implementation support in Prism-MW.
  - Developed iterative redundancy, an algorithm for reliably distributing and replicating computation on untrusted networks of faulty and malicious nodes.
  - Contributed as an architect and developer to a diverse set of experimental software systems, such as computer simulations of volunteer computing networks and collaboration tools for geographically distributed teams of engineers.

- **Research Associate** **May 2009 – Jan 2010**
  Intelligent Systems Technology, Inc., Los Angeles, CA
  - Designed and implemented an instructional game for US Army vehicle maintenance technicians that teaches techniques for making decisions in the presence of uncertainty.
  - Developed research grant proposals for DoD and DHS funding through the Small Business Innovation Research (SBIR) program and Broad Agency Announcements (BAAs).
  - Briefed managers of research funding programs at DoD and DHS on critical research challenges and opportunities in computer science and software engineering.

- **Graduate Research Intern** **May 2008 – Aug 2008**
  T.J. Watson Research Center, IBM, Yorktown Heights, NY
  - Developed elements of BlueStar, a scalable mobile device management and provisioning system. BlueStar allows system administrators to easily manage the configuration of thousands of mobile devices.
  - Co-designed Proxima, a context-aware search service for mobile devices. Proxima provides image-based search capabilities for finding people in a large database by leveraging contextual information associated with an image, such as location, time, and the unique characteristics of the user.
  - Developed requirements for a mobile application for insurance adjusters and managed a team of overseas programmers.

- **Software Engineer** **Jun 2005 – Aug 2006**
  The Boeing Company, Huntington Beach, CA
  - Advised, coordinated, and assisted C4ISR software architecture modeling and analysis efforts leveraging state-of-the-art model-driven engineering technologies.
  - Developed the architecture for a mobile (handheld) device for controlling unmanned autonomous air and ground robots.
  - Integrated multiple independent simulations of interactions among military vehicles and networks to create test harnesses for subcontractor-supplied software components.
  - Created and evaluated software architecture descriptions, guidance memoranda, requirements specifications, design models, trade studies, risk analyses, and development schedules.

- **Graduate Research Assistant** **Sept 2003 – May 2004**
  Computer Science Department, Vanderbilt University, Nashville, TN
  - Developed a CORBA Component Model framework that supports application access to highly optimized and scalable event-based (publish-subscribe) communication services.
  - Created a specification language and supporting toolset for configuring QoS policies in component-based DRE systems.
  - Designed regression tests and enhanced support for key capabilities in a real-time CORBA implementation.

- **Network Technician** **Oct 2002 – Mar 2003**
  CTC Networks, Nashville, TN
  - Administered Windows-based small business networks and performed network and PC upgrades, troubleshooting, and repair.
  - Advised business clients on software purchasing and technology infrastructure investment decisions.

- **e-Business Consultant** **May 2001 – Aug 2001**
  Nebraska e-Commerce Association, Lincoln, NE
  - Advised small businesses on e-commerce development and implementation strategies.

- **Web Developer** **May 1998 – Aug 2000**
  Nebraska Educational Telecommunications, Lincoln, NE
  - Developed front-end HTML and JavaScript and secure grade assessment software for CLASS, the first web-delivered, fully accredited high school.
  - Implemented a multimedia publishing system for web-based distribution of television programs.

LITIGATION CONSULTING

- **Autoscan GmbH et al v 5121175 Manitoba LTD d/b/a DCC Hail** **Nov 2020 – Present**
  - Jurisdiction: Federal Court of Canada
  - Case Number: T-359-20
  - Counsel: Gowling WLG
  - Nature of Suit: Copyright

- **Intellicad Technology Consortium v. <u>Suzhou Gstarsoft Co. Ltd.</u>** **July 2020 – Oct 2020**
  - Jurisdiction: U.S. District Court for the District of Oregon
  - Case Number: 3:19-cv-01963
  - Counsel: Perkins Coie LLP
  - Nature of Suit: Copyright

- **<u>Denika Terry, et al.</u> v. Wasatch Advantage Group, LLC et al.** **June 2020 – July 2020**
  - Jurisdiction: Superior Court of California, County of Los Angeles
  - Case Number: 34-2014-00158417
  - Counsel: Goldstein, Borgen, Dardarian & Ho
  - Nature of Suit: Class Action

- **American National Manufacturing, Inc. v. <u>Sleep Number Corporation</u>** **Sept 2019 – July 2020**
  - Jurisdiction: United States Patent and Trademark Office
  - Case Number: IPR2019-00497, IPR2019-00500, IPR2019-00514
  - Counsel: Fox Rothschild LLP
  - Nature of Suit: Patent

- **Bookman v. <u>Resorts World Casino New York City, Genting New York LLC and International Game Technology PLC</u>** **Sept 2019 – Oct 2019**
  - Jurisdiction: Supreme Court of the State of New York, Queens County
  - Case Number: 0713839/2017
  - Counsel: Proskauer Rose LLP
  - Nature of Suit: Negligence and Breach of Contract

- **<u>IPA Technologies, Inc.</u> v. Amazon.com, Inc. et al.** **May 2019 – Present**
  <u>IPA Technologies, Inc.</u> v. Google LLC
  - Jurisdiction: U.S. District Court for the District of Delaware
  - Case Number: 16-cv-01266, 18-cv-00318
  - Counsel: Skiermont Derby LLP
  - Nature of Suit: Patent

- **<u>Sequoia Technology LLC</u> v. Dell, Inc., Dell Technologies Inc.,** **Oct 2018 – Present**
  Hewlett Packard Enterprise Co., Hitachi Ltd, and Super Micro Computer
  - Jurisdiction: U.S. District Court for the District of Delaware
  - Case Number: 1:18-cv-01127
  - Counsel: One LLP
  - Nature of Suit: Patent

- **Lithography Machines and Systems, and Components Thereof** **Sept 2018 – Apr 2019**
  - Jurisdiction: U.S. International Trade Commission
  - Case Number: 337-TA-1128, 337-TA-1129
  - Counsel: Morrison & Foerster LLP (Nikon Corporation)
  - Nature of Suit: Patent

- **Sleep Number Corporation v. Sizewise Rentals, LLC and American National Manufacturing, Inc.** **Aug 2018 – Present**
  - Jurisdiction: U.S. District Court for the Central District of California
  - Case Number: 5:18-cv-00356
  - Counsel: Fox Rothschild LLP
  - Nature of Suit: Patent

- **Capstone Logistics Holdings, Inc., et al. v. Humano, LLC, et al.** **June 2018 – Jan 2019**
  - Jurisdiction: U.S. District Court for the Southern District of New York
  - Case Number: 1:17-cv-04819
  - Counsel: Ferber Law
  - Nature of Suit: Copyright and Trade Secret

- **Arlynn Whittaker v. Glaser Weil Fink Howard Avchen & Shapiro, LLP** **Mar 2018 – Oct 2018**
  - Jurisdiction: Superior Court of California, County of Los Angeles
  - Case Number: BC657065, BC664506
  - Counsel: Baker Marquart
  - Nature of Suit: Professional Negligence

- **CyWee Group Ltd. v. ZTE Corporation, ZTE (USA), Inc., and ZTE (TX), Inc.** **Feb 2018 – Jan 2019**
  - Jurisdiction: U.S. District Court for the Southern District of California
  - Case Number: 3:17-cv-02130
  - Counsel: Pillsbury Winthrop Shaw Pittman LLP
  - Nature of Suit: Patent

- **SEVEN Networks, LLC v. ZTE (USA) Inc. and ZTE Corporation** **Jan 2018 – Aug 2019**
  - Jurisdiction: U.S. District Court for the Northern District of Texas
  - Case Number: 3:17-cv-01495
  - Counsel: Pillsbury Winthrop Shaw Pittman LLP
  - Nature of Suit: Patent

- **Farmobile, LLC v. Farmers Edge Inc.** **Nov 2017 – Present**
  - Jurisdiction: Federal Court of Canada
  - Case Number: FC 915, FC 688, FC 915, FC 1269
  - Counsel: Gowling WLG
  - Nature of Suit: Patent

- **Space Data Corporation v. Alphabet Inc., Google LLC, and Loon, LLC** **July 2017 – Sept 2019**
  - Jurisdiction: U.S. District Court for the Northern District of California
  - Case Number: 5:16-cv-03260
  - Counsel: Hosie Rice LLP
  - Nature of Suit: Patent and Trade Secret

- **Hitachi Maxell, Ltd. v. ZTE Corp. and ZTE USA Inc.** **July 2017 – June 2018**
  - Jurisdiction: U.S. District Court for the Eastern District of Texas
  - Case Number: 16-cv-00179
  - Counsel: Pillsbury Winthrop Shaw Pittman LLP
  - Nature of Suit: Patent

- **Semcon IP Inc. v. ZTE Corporation, et al.** **May 2017 – Mar 2018**
  - Jurisdiction: U.S. District Court for the Eastern District of Texas
  - Case Number: 16-cv-00437
  - Counsel: Pillsbury Winthrop Shaw Pittman LLP
  - Nature of Suit: Patent

- **Vesta Corporation v. Amdocs Management Limited and Amdocs, Inc.** **Mar 2017 – Nov 2018**
  - Jurisdiction: U.S. District Court for the District of Oregon
  - Case Number: 3:14-cv-01142
  - Counsel: Perkins Coie LLP
  - Nature of Suit: Trade Secret

- **eBay, Inc. v. Masterobjects, Inc.** **Mar 2017 – Jul 2018**
  - Jurisdiction: United States Patent and Trademark Office
  - Case Number: IPR2017-00740
  - Counsel: Carr & Ferrell LLP
  - Nature of Suit: Patent

- **Implicit, LLC v. Trend Micro, Inc.** **Jan 2017 – Oct 2017**
  - Jurisdiction: U.S. District Court for the Eastern District of Texas
  - Case Number: 6:16-cv-00080, 6:17-cv-00183
  - Counsel: DiNovo Price Ellwanger LLP
  - Nature of Suit: Patent

- **Implicit, LLC v. Ericsson, Inc.** **Jan 2017 – Mar 2017**
  - Jurisdiction: U.S. District Court for the Eastern District of Texas
  - Case Number: 6:16-cv-00075, 6:17-cv-00181
  - Counsel: DiNovo Price Ellwanger LLP
  - Nature of Suit: Patent

- **Sound View Innovations, LLC v. Facebook, Inc.** **Aug 2016 – Jan 2018**
  - Jurisdiction: U.S. District Court for the District of Delaware
  - Case Number: 1:16-cv-00116
  - Counsel: Desmarais LLP
  - Nature of Suit: Patent

- **Farmers Edge, Inc., et al. v. Farmobile LLC, et al.** **Jul 2016 – May 2018**
  - Jurisdiction: U.S. District Court for the District of Nebraska
  - Case Number: 8:17-cv-00225
  - Counsel: Husch Blackwell LLP
  - Nature of Suit: Trade Secret and Breach of Contract

- **Certain Air Mattress Systems, Components Thereof, and Methods of Using the Same** **Apr 2016 – May 2016**
  - Jurisdiction: U.S. International Trade Commission
  - Case Number: 337-TA-971
  - Counsel: Pillsbury Winthrop Shaw Pittman LLP (Sleep Number Corporation)
  - Nature of Suit: Patent

- **Synca Direct Inc. v. Scil Animal Care Company and Vet Novations, Inc.** **Mar 2016 – Dec 2016**
  - Jurisdiction: U.S. District Court for the Northern District of New York
  - Case Number: 8:15-cv-00794
  - Counsel: White and Williams LLP
  - Nature of Suit: Breach of Contract

- **TiVo, Inc. v. Samsung Electronics Co., Ltd., et al.** **Mar 2016 – Nov 2016**
  - Jurisdiction: U.S. District Court for the Eastern District of Texas
  - Case Number: 2:15-cv-01503
  - Counsel: Irell & Manella LLP
  - Nature of Suit: Patent

- **Backflip Software, Inc. v. Cisco Systems, Inc.** **Feb 2016 – Oct 2016**
  - Jurisdiction: Superior Court of California, County of Santa Clara
  - Case Number: 2013-1-cv-242234
  - Counsel: Hosie Rice LLP
  - Nature of Suit: Breach of Contract

- **Cellular Communications Equipment, LLC v. HTC Corporation and ZTE Corp.** **Dec 2015 – Aug 2018**
  - Jurisdiction: U.S. District Court for the Eastern District of Texas
  - Case Number: 2:15-cv-00576
  - Counsel: Pillsbury Winthrop Shaw Pittman LLP
  - Nature of Suit: Patent

- **Ericsson Inc., et al. v TCL Communication Technology Holdings, Ltd., et al.** **Sep 2015 – Dec 2015**
  - Jurisdiction: U.S. District Court for the Eastern District of Texas
  - Case Number: 2:15-cv-00011
  - Counsel: Sheppard Mullin Richter & Hampton LLP
  - Nature of Suit: Patent

- **Chipp'd Ltd. v. Crush & Lovely LLC** **Jul 2015 – Sep 2015**
  - Jurisdiction: American Arbitration Association
  - Counsel: White and Williams LLP
  - Nature of Suit: Breach of Contract

- **Bryndon Fisher v. The United States of America** **Aug 2015 – Present**
  - Jurisdiction: United States Court of Federal Claims
  - Case Number: 13-608C
  - Counsel: Schubert Jonckheer & Kolbe LLP
  - Nature of Suit: Class Action

- **Seymore Levine v. The Boeing Company** — **Jul 2015 – Jan 2016**
  - Jurisdiction: U.S. District Court for the Central District of California
  - Case Number: 2:14-cv-06859
  - Counsel: Quinn Emanuel Urquhart & Sullivan LLP
  - Nature of Suit: Patent

- **Global Cash Access, Inc. v. NRT Technology Corp. et al.** — **Jul 2015 – Mar 2016**
  - Jurisdiction: U.S. District Court for the District of Nevada
  - Case Number: 2:15-cv-00822
  - Counsel: Pillsbury Winthrop Shaw Pittman LLP
  - Nature of Suit: Patent

- **Anthony Johnson v. Storix, Inc.** — **Jun 2015 – Dec 2015**
  - Jurisdiction: U.S. District Court for the Southern District of California
  - Case Number: 3:14-cv-01873
  - Counsel: Eastman & McCartney LLP
  - Nature of Suit: Copyright

- **Custom Media Technologies LLC v. AT&T Services, Inc., et al.** — **Apr 2015 – Jun 2015**
  - Jurisdiction: U.S. District Court for the District of Delaware
  - Case Number: 1:13-cv-01419
  - Counsel: Freitas Angell & Weinberg LLP
  - Nature of Suit: Patent

- **Huricks et al. v. Shopkick, Inc.** — **Apr 2015 – Aug 2015**
  - Jurisdiction: U.S. District Court for the Northern District of California
  - Case Number: 4:14-cv-02464
  - Counsel: Newman DuWors, LLP
  - Nature of Suit: Telephone Consumer Protection Act (TCPA)

- **Fox Television Stations, Inc. v. FilmOn X, LLC, et al.** — **Mar 2015 – Jul 2015**
  - Jurisdiction: U.S. District Court for the District of Columbia
  - Case Number: 1:13-cv-00758
  - Counsel: Baker Marquart LLP
  - Nature of Suit: Copyright

- **e-Watch, Inc. et al. v. ZTE Solutions, Inc., ZTE (USA) Inc. and ZTE Corp.** — **Mar 2015 – Jan 2017**
  - Jurisdiction: U.S. District Court for the Eastern District of Texas
  - Case Number: 2:13-cv-01071
  - Counsel: Pillsbury Winthrop Shaw Pittman LLP
  - Nature of Suit: Patent

- **Novatel Wireless, Inc., et al. v. ZTE Corp. and ZTE (USA) Inc.** — **Jan 2015 – Jul 2015**
  - Jurisdiction: U.S. District Court for the Southern District of California
  - Case Number: 3:12-cv-02576
  - Counsel: Pillsbury Winthrop Shaw Pittman LLP
  - Nature of Suit: Patent

- **Essociate, Inc. v. Clickbooth.com, LLC** **Oct 2014 – Feb 2015**
  - Jurisdiction: U.S. District Court for the Central District of California
  - Case Number: 8:13-cv-01886
  - Counsel: Newman DuWors, LLP
  - Nature of Suit: Patent

- **Adaptix, Inc. v. ZTE Corporation, ZTE Solutions, Inc., and ZTE (USA) Inc.** **Sep 2014 – Jul 2016**
  - Jurisdiction: U.S. District Court for the Eastern District of Texas
  - Case Number: 6:13-cv-00438
  - Counsel: Pillsbury Winthrop Shaw Pittman LLP
  - Nature of Suit: Patent

- **VStream Technologies, LLC v. ZTE Corporation and ZTE (USA) Inc., et al.** **Sep 2014 – Jul 2015**
  - Jurisdiction: U.S. District Court for the Eastern District of Texas
  - Case Number: 6:14-cv-00296
  - Counsel: Pillsbury Winthrop Shaw Pittman LLP
  - Nature of Suit: Patent

- **State of California v. Burlena King** **Aug 2014 – Sep 2014**
  - Jurisdiction: Superior Court of California, County of Los Angeles
  - Counsel: Law Offices of the Los Angeles County Public Defender
  - Nature of Suit: Criminal

- **Hill-Rom Company, Inc., et al. v. General Electric Company** **Jul 2014 – Sep 2014**
  - Jurisdiction: U.S. District Court for the Eastern District of Virginia
  - Case Number: 2:14-cv-00187
  - Counsel: Schiff Hardin LLP
  - Nature of Suit: Patent

- **Personal Audio LLC v. ZTE Corporation, et al.** **Jul 2014 – May 2015**
  - Jurisdiction: U.S. District Court for the Eastern District of Texas
  - Case Number: 1:14-cv-00008
  - Counsel: Pillsbury Winthrop Shaw Pittman LLP
  - Nature of Suit: Patent

- **Yardi Systems, Inc. v. Property Solutions International, Inc.** **May 2014 – Sept 2019**
  - Jurisdiction: U.S. District Court for the Central District of California
  - Case Number: 2:13-cv-07764
  - Counsel: Morrison & Foerster, LLP
  - Nature of Suit: Copyright and Trade Secret

- **AgJunction, LLC v. Agrian, Inc. et al.** **Apr 2014 – Feb 2015**
  - Jurisdiction: U.S. District Court for the District of Kansas
  - Case Number: 2:14-cv-02069
  - Counsel: Husch Blackwell LLP
  - Nature of Suit: Copyright and Trade Secret

- **<u>Davies</u> v. L.A. Checker Cab** — **Apr 2014 – Jun 2014**
  - Jurisdiction: Superior Court of California, County of Los Angeles
  - Case Number: SC111290
  - Counsel: Cheong, Denove, Rowell & Bennett
  - Nature of Suit: Negligence

- ***Inter Partes* Review of U.S. Patent No. 8,466,795** — **Feb 2014 – Jul 2014**
  - Jurisdiction: U.S. Patent and Trademark Office
  - Case Number: IPR2014-00670
  - Counsel: Pillsbury Winthrop Shaw Pittman LLP (ZTE Corporation)
  - Nature of Suit: Patent

- **Certain Wireless Devices, Including Mobile Phones And Tablets II** — **Feb 2014 – Jul 2014**
  - Jurisdiction: U.S. International Trade Commission
  - Case Number: 337-TA-905
  - Counsel: Pillsbury Winthrop Shaw Pittman LLP (ZTE Corporation)
  - Nature of Suit: Patent

- **Pragmatus Mobile, LLC v. <u>ZTE Corporation</u>** — **Feb 2014 – Jul 2014**
  - Jurisdiction: U.S. District Court for the District of Delaware
  - Case Number: 1:14-cv-00443
  - Counsel: Pillsbury Winthrop Shaw Pittman LLP
  - Nature of Suit: Patent

- **<u>Intellectual Ventures, LLC</u> v. AT&T Mobility, LLC, et al.** — **Oct 2013 – May 2015**
  <u>Intellectual Ventures, LLC</u> v. T-Mobile USA, Inc., et al.
  <u>Intellectual Ventures, LLC</u> v. Nextel Operations, Inc., et al.
  <u>Intellectual Ventures, LLC</u> v. United States Cellular Corporation
  - Jurisdiction: U.S. District Court for the District of Delaware
  - Case Number: 1:12-cv-00193, 1:13-cv-01631, 1:13-cv-01636, 1:13-cv-01637, 1:13-cv-01634, 1:13-cv-01635, 1:13-cv-01632, 1:13-cv-01633
  - Counsel: Dechert LLP
  - Nature of Suit: Patent

- **<u>Front Row Technologies, LLC</u> v. NBA Media Ventures, LLC, et al.** — **Oct 2013 – July 2016**
  - Jurisdiction: U.S. District Court for the District of New Mexico
  - Case Number: 1:13-cv-01153
  - Counsel: Shore Chan DePumpo LLP
  - Nature of Suit: Patent

- **<u>Nokia Corporation and Nokia Inc.</u> v. HTC Corporation and HTC America, Inc.** — **Dec 2013 – Feb 2014**
  - Jurisdiction: U.S. District Court for the District of Delaware
  - Case Number: 1:12-cv-00549, 1:12-cv-00550, 1:12-cv-00551
  - Counsel: Desmarais LLP
  - Nature of Suit: Patent

- **Certain Portable Electronic Communications Devices, Including Mobile Phones and Components Thereof** **Aug 2013 – Feb 2014**
  - Jurisdiction: U.S. International Trade Commission
  - Case Number: 337-TA-885
  - Counsel: Desmarais LLP (Nokia Corp.)
  - Nature of Suit: Patent

- **SecurityBase.com v. Jeffrey Essick, et al.** **Aug 2013 – Jun 2015**
  - Jurisdiction: Superior Court of California, County of Orange
  - Case Number: 30-2012-00546106
  - Counsel: Younesi & Yoss, LLP
  - Nature of Suit: Copyright and Trade Secret

- **Nokia Corp. and Intellisync Corp. v. HTC America, Inc., and Exedea, Inc.** **Feb 2013 – May 2013**
  - Jurisdiction: U.S. District Court for the District of Delaware
  - Case Number: 1:12-cv-00549, 1:12-cv-00550
  - Counsel: Desmarais LLP
  - Nature of Suit: Patent

- **Certain Electronic Devices, Including Mobile Phones and Tablet Computers, and Components Thereof** **Feb 2013 – May 2013**
  - Jurisdiction: U.S. International Trade Commission
  - Case Number: 337-TA-847
  - Counsel: Desmarais LLP (Nokia Corp.)
  - Nature of Suit: Patent

- **Essociate, Inc. v. Azoogle.com, Inc. and Epic Media Group, Inc.** **Nov 2012 – Oct 2013**
  - Jurisdiction: U.S. District Court for the Western District of Wisconsin
  - Case Number: 3:11-cv-00727
  - Counsel: Newman DuWors, LLP
  - Nature of Suit: Patent

- **John Doe v. Passageway School** **Oct 2012 – Dec 2012**
  - Jurisdiction: Superior Court of California, County of Ventura
  - Case Number: 2012-00418623
  - Counsel: Stone & Hiles, LLP
  - Nature of Suit: Negligence

- **Essociate, Inc. v. Neverblue Media, Inc.** **Sep 2012 – Jul 2013**
  - Jurisdiction: U.S. District Court for the Central District of California
  - Case Number: 2:12-cv-08455
  - Counsel: Newman DuWors, LLP
  - Nature of Suit: Patent

- **Porto Technology Co., Ltd. v. Cellco Partnership d/b/a Verizon Wireless** **Sep 2012 – Feb 2014**
  - Jurisdiction: U.S. District Court for the Eastern District of Virginia
  - Case Number: 3:12-cv-00678, 3:13-cv-00265
  - Counsel: Fox Rothschild LLP
  - Nature of Suit: Patent

- **Flashpoint Technology, Inc. v. ZTE Corporation** **Aug 2012 – Mar 2013**
  - Jurisdiction: U.S District Court for the District of Delaware
  - Case Number: 12-cv-00649
  - Counsel: Goodwin Procter LLP
  - Nature of Suit: Patent

- **Certain Electronic Imaging Devices** **Aug 2012 – Mar 2013**
  - Jurisdiction: U.S International Trade Commission
  - Case Number: 337-TA-850 (ZTE Corporation)
  - Counsel: Goodwin Procter LLP
  - Nature of Suit: Patent

- **MobileMedia Ideas, LLC v. HTC Corporation and HTC America, Inc.** **Jul 2012 – May 2013**
  - Jurisdiction: U.S. District Court for the Eastern District of Texas
  - Case Number: 2:10-cv-00112
  - Counsel: Desmarais LLP
  - Nature of Suit: Patent

- **Golden Archer Investments, LLC v. SkyNet Financial Systems, LLC** **Jul 2012 – Dec 2012**
  - Jurisdiction: U.S. District Court for the Southern District of New York
  - Case Number: 1:11-cv-03673
  - Counsel: Goldman Ismail Tomaselli Brennan & Baum LLP
  - Nature of Suit: Breach of Contract

- **Inter Partes Re-examination of U.S. Patent No. 7,702,669** **May 2012 – Jul 2012**
  - Jurisdiction: U.S. Patent and Trademark Office
  - Case Number: 95/001,844
  - Counsel: Hickman Palermo Truong Becker Bingham Wong LLP (RingCentral, Inc.)
  - Nature of Suit: Patent

- **Denise Black v. Prowess, Inc.** **Mar 2012 – May 2012**
  - Jurisdiction: State of New York, Supreme Court, County of Ontario
  - Case Number: 0102070/2009
  - Counsel: Smith, Sovik, Kendrick & Sugnet, PC
  - Nature of Suit: Breach of Contract

- **Essociate, Inc. v. Direct ROI, LLC** **Mar 2012 – Jun 2012**
  - Jurisdiction: U.S. District Court for the Central District of California
  - Case Number: 2:10-cv-02107, 8:12-cv-00444
  - Counsel: Newman DuWors, LLP
  - Nature of Suit: Patent

- **<u>EchoStar Technologies, Corp.</u> v. TiVo Inc. and Humax USA, Inc.** **Mar 2011 – Apr 2011**
  - Jurisdiction: U.S. District Court for the Eastern District of Texas
  - Case Number: 5:05-cv-00081
  - Counsel: Morrison & Foerster, LLP
  - Nature of Suit: Patent

JOURNAL AND MAGAZINE ARTICLES

- Mahdi Eslamimehr, George Edwards, and Mohsen Lesani, ***Efficient Detection and Validation of Atomicity Violations in Concurrent Programs***, Journal of Systems & Software, 2017.

- Yuriy Brun, Jae young Bang, George Edwards, and Nenad Medvidovic, ***Self-Adapting Reliability in Distributed Software Systems***, IEEE Transactions on Software Engineering, 2015.

- Chris A. Mattmann, Nenad Medvidovic, Sam Malek, George Edwards and Somo Banerjee, ***A Middleware Platform for Providing Mobile and Embedded Computing Instruction to Software Engineering Students***, IEEE Transactions on Education, 2012.

- Nenad Medvidovic, Hossein Tajalli, Joshua Garcia, Yuriy Brun, Ivo Krka, and George Edwards, ***Engineering Heterogeneous Robotics Systems: A Software Architecture-Based Approach***, IEEE Computer, 2011.

- Nenad Medvidovic and George Edwards, ***Software Architecture and Mobility: A Roadmap***, Journal of Systems and Software (JSS): Special Issue on Software Architecture and Mobility, 2010.

- Nenad Medvidovic, Hossein Tajalli, Joshua Garcia, Yuriy Brun, Ivo Krka, George Edwards, Marija Mikic-Rakic, Sam Malek, and Gaurav Sukhatme, ***An Architecture-Driven Software Mobility Framework***, Journal of Systems and Software (JSS): Special Issue on Software Architecture and Mobility, 2010.

- George Edwards, Chiyoung Seo, and Nenad Medvidovic, ***Model Interpreter Frameworks: A Foundation for the Analysis of Domain-Specific Software Architectures***, Journal of Universal Computer Science (JUCS): Special Issue on Software Components, Architectures and Reuse, 2008.

- Aniruddha Gokhale, Krishnakumar Balasubramanian, Jaiganesh Balasubramanian, Arvind Krishna, George T. Edwards, Gan Deng, Emre Turkay, Jeffrey Parsons, and Douglas C. Schmidt, ***Model-Driven Middleware: A New Paradigm for Deploying and Provisioning Distributed Real-time and Embedded Applications***, Elsevier Journal of the Science of Computer Programming: Special Issue on Model Driven Architecture, 2005.

CONFERENCE PAPERS

- Mahdi Eslamimehr and George Edwards, ***End-to-End Cross-Language Test Case Generation for Web Applications***, Proceedings of the 9th EAI International Conference on Performance Evaluation Methodologies and Tools (VALUETOOLS), December 2015.

- Christoph Dorn, George Edwards, and Nenad Medvidovic, ***Analyzing Design Tradeoffs in Large-scale Socio-Technical Systems through Simulation of Dynamic Collaboration Patterns***, Proceedings of the 20th International Conference on Cooperative Information Systems (CoopIS), September 2012.

- George Edwards, Nenad Medvidovic, and Yuriy Brun, ***Automated Analysis and Code Generation for Domain-Specific Models***, Proceedings of the Joint 10th Working IEEE/IFIP Conference on Software Architecture & 6th European Conference on Software Architecture (WICSA/ECSA), August 2012.

- George Edwards, Yuriy Brun, and Nenad Medvidovic, ***Isomorphism in Model Tools and Editors***, Proceedings of the 26th IEEE/ACM International Conference on Automated Software Engineering (ASE), November 2011.

- Yuriy Brun, George Edwards, Jae young Bang and Nenad Medvidovic, ***Smart Redundancy for Distributed Computation***, Proceedings of 31st International Conference on Distributed Computing Systems (ICDCS), June 2011.

- Hossein Tajalli, Joshua Garcia, George Edwards, and Nenad Medvidovic, ***PLASMA: A Plan-based Layered Architecture for Software Model-driven Adaptation***, Proceedings of the 25th IEEE/ACM International Conference on Automated Software Engineering (ASE), September 2010.

- Jae young Bang, Daniel Popescu, George Edwards, Nenad Medvidovic, Naveen Kulkarni, Girish M. Rama, and Srinivas Padmanabhuni, ***CoDesign - A Highly Extensible Collaborative Software Modeling Framework***, Research Demonstrations Track, Proceedings of the ACM/IEEE 32nd International Conference on Software Engineering (ICSE), May 2010.

- Ivo Krka, George Edwards, Yuriy Brun and Nenad Medvidovic, ***Synthesizing Partial Component-Level Behavior Models from System Specifications***, Proceedings of the 7th Joint Meeting of the European Software Engineering Conference and ACM SIGSOFT Symposium on the Foundations of Software Engineering (ESEC/FSE), August 2009.

- Joshua Garcia, Daniel Popescu, George Edwards and Nenad Medvidovic, ***Toward a Catalogue of Architectural Bad Smells***, Proceedings of the 5th International Conference on the Quality of Software Architectures (QoSA), June 2009.

- Ivo Krka, George Edwards, Yuriy Brun and Nenad Medvidovic, ***From System Specifications to Component Behavioral Models***, New Ideas and Emerging Results Track, Companion Volume of the 31st International Conference on Software Engineering (ICSE), May 2009.

- Joshua Garcia, Daniel Popescu, George Edwards and Nenad Medvidovic, ***Identifying Architectural Bad Smells***, Proceedings of the 13th European Conference on Software Maintenance and Reengineering (CSMR), March 2009.

- Steve Mastrianni, David Bantz, Terrence Buechner, Tom Chefalas, George Edwards, Song Jinho, Dong Jun Lan, Gary Leonardi, Leslie Liu, Randy Moulic, Dennis G. Shea, Andrew Wyskida, ***BlueStar: Managed Services for Enterprise Mobility***, Proceedings of the IEEE International Conference on e-Business Engineering (ICEBE), October 2008.

- George Edwards, Leslie S. Liu, and Randy Moulic, ***Proxima: A Mobile Augmented-Image Search System***, Proceedings of the ACM International Conference on Multimedia (ACMMM), October 2008.

- George Edwards and Nenad Medvidovic, ***A Methodology and Framework for Creating Domain-Specific Development Infrastructures***, Proceedings of the 23rd IEEE ACM International Conference on Automated Software Engineering (ASE), September 2008.

- Chiyoung Seo, George Edwards, Sam Malek and Nenad Medvidovic, ***A Framework for Estimating the Impact of a Distributed Software System's Architectural Style on its Energy Consumption***, Proceedings of the Working International Conference on Software Architecture (WICSA), February 2008.

- George Edwards, Chiyoung Seo, and Nenad Medvidovic, ***Construction of Analytic Frameworks for Component-Based Architectures***, Proceedings of the Brazilian Symposium on Software Components, Architectures and Reuse (SBCARS), August 2007.

- Gan Deng, Ming Xiong, Aniruddha Gokhale, and George Edwards, ***Evaluating Real-time Publish/Subscribe Service Integration Approaches in QoS-enabled Component Middleware***, Proceedings of the 10th IEEE International Symposium on Object-oriented Real-time Distributed Computing (ISORC), May 2007.

- George Edwards, Sam Malek, and Nenad Medvidovic, ***Scenario-Driven Dynamic Analysis of Distributed Architectures***, Proceedings of the 10th International Conference on Fundamental Approaches to Software Engineering (FASE), March 2007.

- George Edwards, Gan Deng, Douglas C. Schmidt, Anirudda Gokhale, and Balachandran Natarajan, ***Model-Driven Configuration and Deployment of Component Middleware Publisher/Subscriber Services***, Proceedings of the 3rd ACM International Conference on Generative Programming and Component Engineering (GPCE), October 2004.

- George Edwards, Douglas C. Schmidt, Aniruddha Gokhale, and Bala Natarajan, ***Integrating Publisher/Subscriber Services in Component Middleware for Distributed Real-time and Embedded Systems***, Proceedings of the 42nd Annual ACM Southeast Conference (ACMSE), April 2004.

WORKSHOP PAPERS

- Arman Shahbazian, George Edwards and Nenad Medvidovic, ***An End-to-End Domain Specific Modeling and Analysis Platform***, 8th Workshop on Modelling in Software Engineering (MiSE), May 2016.

- Chiyoung Seo, George Edwards, Daniel Popescu, Sam Malek and Nenad Medvidovic, ***A Framework for Estimating the Energy Consumption Induced by a Distributed System's Architectural Style***, 8th Workshop on Specification and Verification of Component-Based Systems (SAVCBS), August 2009.

- George Edwards, Joshua Garcia, Hossein Tajalli, Daniel Popescu, Nenad Medvidovic, Gaurav Sukhatme, and Brad Petrus, ***Architecture-Driven Self-Adaptation and Self-Management in Robotics Systems***, Proceedings of the Workshop on Software Engineering for Adaptive and Self-Managing Systems (SEAMS), May 2009.

- Ivo Krka, Leslie Cheung, George Edwards, Leana Golubchik and Nenad Medvidovic, ***Architecture-Based Software Reliability Estimation: Problem Space, Challenges, and Strategies***, Proceedings of the Workshop on Architecting Dependable Systems (WADS), June 2008.

- George Edwards, Chiyoung Seo, Daniel Popescu, Sam Malek, and Nenad Medvidovic, ***Self-* Software Architectures and Component Middleware in Pervasive Environments***, Proceedings of the 5th International Workshop on Middleware for Pervasive and Ad-Hoc Computing (MPAC), November 2007.

- Chiyoung Seo, Sam Malek, George Edwards, Daniel Popescu, Nenad Medvidovic, Brad Petrus, and Sharmila Ravula, ***Exploring the Role of Software Architecture in Fault-Tolerant and Dynamic Pervasive Systems***, Proc. of 1st Workshop on Software Engineering of Pervasive Computing Applications, Systems and Environments (SEPCASE), May 2007.

BOOK CHAPTERS

- Ivo Krka, George Edwards, Leslie Cheung, Leana Golubchik, and Nenad Medvidovic, ***A Comprehensive Exploration of Challenges in Architecture-Based Reliability Estimation***, Architecting Dependable Systems 6, R. de Lemos, J.-C. Fabre, C. Gacek, F. Gadducci, M. ter Beek (Eds.), 2009.

POSTERS AND DEMONSTRATIONS

- ***BlueStar Mobile Management Services***, IBM Research Technology Fair, June 2008.

- ***The Extensible Tool-chain for Evaluation of Architectural Models***, USC Center for Software and Systems Engineering Convocation, October 2006.

- ***A Model-Driven Framework for Architectural Evaluation of Mobile Software Systems***, UCI Institute for Software Research Forum, April 2006.

INVITED TALKS

- ***Metamodeling-Enabled Model-Checking for Complex Systems***, Systems Engineering Research Center (SERC) Annual Research Review, November 2010.

- ***XTEAM: Automated Synthesis of Domain-Specific Code Generators***, Northrop Grumman Technology Day Research Showcase, Information Systems Track, May 2010.

- ***Domain-Specific Model Analysis and Code-Generation Frameworks***, Ground Systems Architecture Workshop (GSAW), Architecture-Centric Evolution (ACE) Working Group, March 2010.

- ***Automated Analysis and Code Generation for Domain-Specific Models***, USC Computer Science Department Annual Research Review, March 2010.

- ***Automated Analysis and Code Generation for Domain-Specific Models***, USC Center for Software and Systems Engineering Annual Research Review, March 2010.

- ***Model-Driven Analysis Frameworks for Embedded Systems***, USC Center for Software and Systems Engineering Annual Research Review, March 2009.

- ***Creating Domain-Specific Development Infrastructures***, USC Center for Software and Systems Engineering Annual Research Review, March 2008.

- ***Weighing Architectural Trade-offs with XTEAM***, USC Center for Software and Systems Engineering Annual Research Review, February 2007.

TECHNICAL REPORTS

- George Edwards, Yuriy Brun, and Nenad Medvidovic, ***Automated Analysis and Code Generation for Domain-Specific Models***, Technical Report USC-CSSE-2010-517, Center for Software and Systems Engineering, University of Southern California, August 2010.

- George Edwards and Nenad Medvidovic, ***Model Interpreter Frameworks***, Technical Report USC-CSSE-2009-514, USC Center for Software and Systems Engineering, Center for Software and Systems Engineering, University of Southern California, July 2009.

- George Edwards and Nenad Medvidovic, ***A Highly Extensible Simulation Framework for Domain-Specific Architectures***, Technical Report USC-CSSE-2009-511, Center for Software and Systems Engineering, Univ. of Southern California, May 2009.

- Yuriy Brun, George Edwards, and Nenad Medvidovic, ***Injecting Robustness into Autonomic Grid Systems***, Technical Report USC-CSSE-2009-510, USC Center for Software and Systems Engineering, University of Southern California, May 2009.

PATENT APPLICATIONS

- ***Method and Apparatus for Simulation of Domain-Specific Models***. U.S. Provisional Patent Application 61/513,357, filed July 29, 2011. (abandoned)

- ***Extensible Collaborative Software Modeling***. U.S. Patent Application 13/271,008, filed October 11, 2011. (abandoned)

HONORS AND AWARDS

- **2008 USC Computer Science Outstanding Student Research Award**
  - Awarded each year for exceptional research achievement by a graduate student in computer science.

- **Annenberg Graduate Fellowship**, Aug 2007
  - Supports highly qualified graduate students to conduct cutting-edge communication and digital media research and advance important new programs in the communications arena.

- **USC Viterbi School of Engineering Dean's Doctoral Fellowship**, May 2004
  - Supports world-class students for four years in the pursuit and publication of research leading to the doctoral degree.

- **Vanderbilt *Summa Cum Laude* graduate**, Aug 2003
  - Awarded to graduates whose grade point average equals or exceeds that of the top 5 percent of the previous year's graduating seniors.

- **NSF Research Experiences for Undergraduates Grant**, June 2003
  - Supports active research participation by undergraduate students in any of the areas of research funded by the National Science Foundation.

- **Vanderbilt Dean's List with High Honors**, Dec 2000, May '01, Dec '01, May '02, Dec '02, May '03
  - Recognizes outstanding academic performance in a semester. Students are named to the Dean's List when they earn a grade point average of at least 3.500.

- **Vanderbilt School of Engineering Merit Scholarship**, May 2000
  - Awarded to approximately 5 percent of all applicants based on exceptional accomplishment and high promise in intellectual endeavors.

- **National Merit Scholar**, May 2000
  - Awarded for academic achievement to 8,200 high school students each year out of 1.5 million entrants.

TEACHING

- **Lecturer**, CSCI 102 – Data Structures, Fall 2012

- **Lecturer**, CSCI 568 – Requirements Engineering, Spring 2012, Spring 2013, Spring 2014, Spring 2015, Spring 2016

- **Teaching Assistant**, Infosys Intensive Course on Software Architecture, December 2009.

- **Guest Lecturer**, CSCI 589 – Software Engineering for Embedded Systems, Fall 2007, Fall 2008.

- **Guest Lecturer**, CSCI 377 – Introduction to Software Engineering, Fall 2007, Fall 2008, Fall 2009.

- **Guest Lecturer**, SAE 599 – Model-Driven Systems Architecture, Summer 2007, Spring 2008.

- **Teaching Assistant**, CSCI 589 – Software Engineering for Embedded Systems, Fall 2006.

RESEARCH COMMUNITY SERVICE

- Reviewer for the **Journal of Systems and Software**, Elsevier, 2010 – 2013.

- Reviewer for **IEEE Transactions on Software Engineering (TSE)**, 2010 – 2012.

- Reviewer for the **50th International Conference on Objects, Models, Components, Patterns (TOOLS Europe)**, Prague, Czech Republic, May 28 – June 1, 2012.

- Program Committee Member for the **Research in Applied Computation Symposium (RACS)**, Miami, FL, November 2 – 5, 2011.

- Reviewer for the **Workshop on Software Engineering for Adaptive and Self-Managing Systems (SEAMS), Cape Town**, South Africa, May 3 – 4, 2010.

- Reviewer for the **24th IEEE/ACM International Conference on Automated Software Engineering (ASE)**, Auckland, New Zealand, November 16 – 20, 2009.

- Reviewer for **DSN 2009 Workshop on Architecting Dependable Systems (WADS)**, Lisbon, Portugal, June 29, 2009.

- Reviewer for **Architecting Dependable Systems 6**, Springer Publishing, 2009.

- Reviewer for the **6th IEEE International Conference on Autonomic Computing (ICAC)**, June 15 – 19, 2009.

- Reviewer for **IEEE Software Special Issue: Domain-Specific Languages & Modeling**, July/August, 2009.

- Reviewer for the **23rd International Conference on Automated Software Engineering (ASE)**, L'Aquila, Italy, Sept. 15 – 19, 2008.

- Reviewer for the **Working IEEE/IFIP Conference on Software Architecture (WICSA)**, Vancouver, Canada, February 18 – 21, 2008.

- Program Committee Member for the **Institute for Software Research (ISR) Graduate Student Research Symposium (GSRS)**, Irvine, CA, June 1, 2007.

- Reviewer for the **3rd International Conference on the Quality of Software Architectures (QoSA)**, Boston, MA, July 12 – 13, 2007.

- Reviewer for the **10th International ACM SIGSOFT Symposium on Component-Based Software Engineering (CBSE)**, Boston, MA, July 9 – 11, 2007.

- Reviewer for the **First IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO)**, Boston, MA, July 9 – 11, 2007.

- Reviewer for the **Second Workshop on Sharing and Reusing Architectural Knowledge - Architecture, Rationale, and Design Intent (SHARK/ADI)**, Minneapolis, MN, May 19 – 20, 2007.

- Reviewer for the **Workshop on Tools, Operating Systems and Programming Models for Developing Reliable Systems (TOPMoDelS)**, Long Beach, CA, March 26 – 30, 2007.

PROFESSIONAL ASSOCIATIONS

- **Member**, Association for Computer Machinery (ACM) and ACM Special Interest Group on Simulation and Modeling

- **Member**, IEEE Computer Society

EXHIBIT B

97-7197L

CORRECTED AMICUS BRIEF **97-7327XAD**

IN THE
**UNITED STATES COURT OF APPEALS
FOR THE SECOND CIRCUIT**

Harbor Software, Inc.,

Plaintiff-Appellee-
Cross-Appellant,

-- v. --

Applied Systems, Inc.,

Defendant-Appellant-
Cross-Appellee.

ON APPEAL FROM THE UNITED STATES DISTRICT COURT
FOR THE SOUTHERN DISTRICT OF NEW YORK

AMICUS BRIEF OF COMPUTER SCIENTISTS RE: SOFTWARE COPYRIGHT AND TRADE SECRET CASES -- OBSERVATIONS ON THE ABSTRACTION, FILTRATION AND COMPARISON TEST

Professor Hal Abelson, Professor Roy Campbell,
Professor Randall Davis, Professor Lee Hollaar,
Professor Gerald J. Sussman

Of Counsel:

Marc M. Arkin
140 West 62nd Street
New York, New York 10023
(212) 636 6850
Counsel to Amici Curiae

September 25, 1997

Harold (Hal) Abelson is Class of 1922 Professor of Electrical Engineering and Computer Science at MIT and a Fellow of the IEEE. He received the A.B. degree, summa cum laude, from Princeton University in 1969. In 1973, he received the Ph.D. degree in mathematics from MIT and joined the MIT faculty. In 1992, Abelson was designated as one of MIT's six inaugural MacVicar Faculty Fellows, in recognition of his significant and sustained contributions to teaching and undergraduate education. Abelson was recipient in 1992 of the Bose Award (MIT's School of Engineering teaching award). Abelson is also the winner of the 1995 Taylor L. Booth Education Award given by IEEE Computer Society cited for his continued contributions to the pedagogy and teaching of introductory computer science.

Professor Roy Campbell received his Honors B.S. Degree in Mathematics, with a Minor in Physics from the University of Sussex in 1969 and his M.S. and Ph.D. Degrees in Computer Science from the University of Newcastle upon Tyne in 1972 and 1976, respectively. In 1976 he joined the faculty of the University of Illinois, where he is currently a Full Professor of Computer Science. During the past twenty-one years, he has supervised the completion of twenty-four Ph.D. dissertations and over seventy M.S. theses. He is the author of over one hundred research papers on programming languages, software engineering, operating systems, distributed systems, and networking.

Randall Davis has been on the faculty at MIT since 1978 and is currently a Professor in the Electrical Engineering and Computer Science Department. Dr. Davis has been one of the seminal contributors to the field of artificial intelligence and was selected in 1984 as one of America's top 100 scientists under the age of 40 by Science Digest. In 1986 he received the AI Award from the Boston Computer Society for his contributions to the field. In 1990 he was named a Founding Fellow of the American Association of AI and in 1995 was elected President of the Association. Dr. Davis has been active in the area of intellectual property and software. In 1989 he served as expert to the Court in Computer Associates v. Altai. In 1990 he served as a panelist in a series of workshops on the issue run by the Computer Science and Telecommunications Board of the National Academy of Science, resulting in the publication of Intellectual Property Issues in Software in 1991. He has served as an technical expert in a variety of software cases.

Lee A. Hollaar is a Professor of Computer Science at the University of Utah, where he has taught a variety of computer software and hardware courses and currently teaches intellectual property and computer law. He has been an expert in a number of computer-related cases. While on sabbatical leave during the 1996-97 academic year, he was a Committee Fellow in the intellectual property unit of the Committee on the Judiciary of the United States Senate, advising on computer-related intellectual property issues, and an intern at the United States Court of Appeals for the Federal Circuit. He received

his Ph.D. degree in computer science from the University of Illinois at Urbana-Champaign in 1975 and is a registered patent agent.

Gerald Jay Sussman is the Matsushita Professor of Electrical Engineering at the Masachusetts Institute of Technology. He received the S.B. and the Ph.D. degrees in mathematics from the Massachusetts Institute of Technology in 1968 and 1973, respectively. He has been involved in artificial intelligence research at M.I.T. since 1964. He is a coauthor (with Hal Abelson and Julie Sussman) of the introductory computer science textbook used at M.I.T. The textbook, "Structure and Interpretation of Computer Program," has been translated into French, German, Chinese, and Japanese. As a result of this and other contributions to computer-science education, Sussman received the ACM's Karl Karlstrom Outstanding Educator Award in 1990, and the Amar G. Bose award for teaching in 1991.

Table of Contents

Table of Authorities

Cases **Page(s)**

Statutes

Rules

Miscellaneous

I. INTRODUCTION

This amicus brief is filed on behalf of a group of individual computer scientists[1] who are concerned that there is uncertainty among the courts in how to implement the process suggested by this Court in *Computer Assocs. Int'l., Inc. v. Altai Inc.*, 982 F.2d 693 (2d Cir. 1992) ("*Altai*"), for evaluating copyright infringement claims involving computer software. This uncertainty seems particularly manifest in cases involving allegations of both copyright infringement and misappropriation of trade secrets. The signatories to this brief include a number of pioneering computer scientists who have substantial technical backgrounds and considerable experience as technical experts in intellectual property lawsuits involving computer software. None of the signatories has any relationship -- including a financial relationship -- to any party in the present case. From reading the district court's opinions and the parties' briefs (we have not seen the record which we understand is under seal) we believe that this case illustrates many of the difficulties experienced by trial courts generally in applying the Altai test. Our interest is to assist this Court in providing further guidance to trial courts regarding the process for determining trade secret and copyright in computer software, in a manner that protects the interests of the public, the software industry, and the free flow of ideas.

---

[1] A full list of the amici and their professional qualifications and contributions to the computer software industry precedes the table of contents of this brief.

In Altai, this Court adopted a three-step process of abstraction, filtration, and comparison ("AFC") as the means by which a court should determine the copyright-protectable aspects of a computer program and evaluate claims of copyright infringement. Altai, however, provided little guidance as to precisely when or how the district courts should implement the AFC test and its relevance, if any, to related claims of trade secret misappropriation. In this brief we describe the technical complexity and difficulty of performing the AFC test from the standpoint of computer science, which we have experienced in our role as experts. We offer as well a procedural approach by which this Court could clarify the Altai decision for the benefit of the lower courts. We suggest a procedure for applying the AFC test that we believe will be useful in copyright cases and in cases involving both trade secret and copyright claims. Under this procedure the court would require that, prior to beginning the AFC inquiry, the plaintiff identify its trade secrets with specificity, so that these claims are not affected by the unrelated issues of copyright protection. The court would then proceed with the AFC test for copyright protectability through the evaluation of a focused subset of code identified by plaintiff as alleged to have been copied by defendant, using a process in which experts for both plaintiffs and defendants are able to challenge each other's abstraction exhibits by citing specific technical criteria.

We believe that this approach will minimize the difficulties in implementing the AFC test and will help ensure both that the abstractions are maximally useful to the trial court and that the

entire AFC process proceeds on a technically sound basis. We also believe that our suggested procedural steps will serve the interests of judicial economy by focusing and testing plaintiff's claims at a relatively early stage in the process, while still preserving plaintiff's right to a vigorous prosecution of its case.

We have attempted to keep this brief as jargon-free as possible; as some technical terms were unavoidable, we have included in Appendix II brief definitions of all the terms employed.

Two brief points of non-technical terminology. First, when talking about levels of abstraction, we include the literal text of the program as one of those levels, evidently the lowest level. This inclusion simplifies both the discussion and practice of the AFC process: within one framework we have AFC accomplishing both the levels of the abstraction test envisioned by Judge Hand, and the methodical filtration and comparison of the literal code traditionally carried out by the courts.[2] When discussing a program here we generally proceed from the highest level down to more detailed levels of abstraction, even though the process of constructing the abstractions may start (as Judge Hand suggests) with the literal code. Second, in speaking about software or code, we mean by it all of the things that go into a program: all of the algorithms, data structures, textual commentary, etc.

---

[2] We have written elsewhere on techniques for comparing the literal aspects of software; see, e.g., Davis, R., "The Nature of Software and Its Consequences for Establishing and Evaluating Similarity, Software Law Journal," 299, Vol. V, No. 2, (April 1992).

II. TECHNICAL COMPLEXITIES INHERENT IN THE AFC TEST

Performing the AFC analysis in the context of litigation is a challenging task, in no small measure because of the technical difficulties that arise. These technical difficulties stem from (1) the sheer magnitude of the task of analyzing programs that routinely consist of hundreds of thousands of lines of computer code; (2) the lack of any fixed or agreed-upon set of levels of abstraction by which to describe a program; (3) the interaction of legal doctrines (such as merger, scenes a faire and public domain) with the technical constraints of the computer industry; and (4) the rapid evolution of these doctrines in the area of computer software.

A. Abstracting Software is a Difficult Technical Task Even for an Expert.

The first source of difficulty is size: commercial computer programs often consist of hundreds of thousands of lines of code and can, at times, run to millions of lines.[3] These programs involve a level of complexity that vastly overshadows even the most gargantuan literary work. A technically accurate and precise analysis of so much detailed material is a prodigious undertaking, even for a computer professional.

Additional difficulty arises from the lack of a fixed standard for the levels at which to describe a program. While

---

[3] The latest version of Microsoft's Word program, for example, contains 2.7 million lines of source code.

abstracting software is a familiar concept to software developers, there is as yet no well-defined standard for selecting which levels of abstraction to use in describing a program and no fixed set of levels that properly characterize any given program. While this indefiniteness is commonplace in the technical world, it creates uncertainty and confusion when transferred to the context of the adversary process. Even the Tenth Circuit's effort in Gates Rubber Co. v. Bando Chemical Industries, Ltd., 9 F.3d 823 (10th Cir. 1993), to specify a set of levels of abstraction to use in performing the AFC analysis offers a degree of guidance akin to saying that a play can be described by its characters and plot; it provides a starting point but little more to a person charged with describing a real play.

The difficulty is of course not unique to software: there may be differences in describing the abstract elements of a literary plot depending, for example, on the nature of the work involved and the perspective of the reader. But the problem is more serious in the case of software due to the larger size and complexity of programs as compared to literary works, and due to the relative youth of software as a medium of expression, especially compared to traditional literature with its long history of literary analysis.

B. Filtration Includes a Difficult Technical Task That Can Benefit from Expert Opinion.

A third difficulty arises because the filtration process requires the court to determine whether certain elements of a program fall outside copyright protection because of doctrines

such as merger, scenes a faire, public domain, or whether a body of code does little more than embody a fact about the world. These legal judgments require substantial technical expertise. It may be a challenging task even for a technical expert to determine whether the expression in a body of code is necessarily incidental to the idea being expressed (merger); whether it is dictated by external factors (scenes a faire) such as hardware compatibility (constraints imposed by the computer in use), software compatibility (constraints imposed by software), or industry standards; and whether a body of code is substantially similar to or clearly derived from code in the public domain. Given the difficulty and complexity of making these technical judgments, this Court should clarify that this is properly a task that can benefit from the assistance of a qualified expert in the field. This is made all the more pressing by the fact that the legal doctrines involved in these decisions are themselves rapidly evolving as they apply to computers, further complicating the task and making the assistance of a qualified expert all the more useful.

C. Abstraction and Filtration are Sensible Technical Tasks.

Despite the difficulty of the task, creating a set of abstractions and answering the technical questions raised in filtration are still sensible and well-founded undertakings. As an illustration, two unbiased experts should be able to agree on whether an abstract description of a program was, among other things, complete and technically accurate. While there is judgment involved, there is also a solid body of science

underpinning that judgment. There are, for example, several well-established notions of what kinds of abstractions make sense for a program, including the control structure, data structures, data flow, information architecture, and the textual organization of the code. Each of these is described in more detail in the Appendix I.

III. TRADE SECRETS AS A THRESHOLD ISSUE DISTINCT FROM COPYRIGHT AND THE AFC PROCESS.

A. The Nature of Trade Secrets in Software.

In the present case, the trial court used the abstraction exhibits to address matters of both copyright infringement and trade secret misappropriation. We believe that this demonstrates a significant confusion about the nature of both the abstraction exhibits and the trade secrets likely to be embodied in computer software. Specifically, an abstraction exhibit simply describes part of the program's code; alone, it does not indicate whether an element of code has the economic value necessary to establish its status as a trade secret.

As computer scientists, we would like to suggest that this Court take this opportunity to clarify the relationship between the abstractions prepared to evaluate the copyright protectability of the elements of computer software and the proof necessary to establish that elements of a program are entitled to trade secret protection. As a procedural matter, we would like to suggest that the plaintiff be required at the outset to specify the trade secret, the specific code in which it is embodied, and the alleged economic value of the secret.

We understand that it is common industry practice to maintain the source code of a computer program as a trade secret. Clearly, the literal code of a program may meet the criteria for a trade secret; that is, its owner may have kept it secret and it may have economic value from not being generally known in the industry. The issue is more complicated when the matter in dispute is the trade secret status of non-literal aspects of a program of the type that may be expressed in an abstraction. In claims of this nature, it is vital that the plaintiff specify both the abstraction and an indication of what economic value the abstraction purportedly produces; the abstraction alone is never enough.

As a hypothetical example, consider a program designed to automate a checkbook; among other tasks, it is able to print checks, display your check register, and reconcile the balance at the end of the month. Abstractions used to describe this program might include the organization of the database and the algorithm used for balance reconciliation.

While these abstractions alone may be sufficient information from which to evaluate copyright issues, a trade secret claim must, in addition, specify the economic value each abstraction allegedly adds to the program. Because all of the value of software to the end-user is in its behavior, a specification of value in turn means specifying what economically valuable behavior the abstraction produces.

As an example, a technically substantive trade secret claim might suggest that the design of the database added economic value because it enabled great compactness, i.e., allowed the

program to store a very large number of checks in very little space (and this had worth in the marketplace).

Without specification of the valuable behavior contributed by an element of a program, a trade secret claim cannot be sensibly evaluated from a technical point of view. Consider the design of the database: to ask whether the design is one that would result from information generally known in the field, we have to know what goal, i.e., what economically valuable behavior, the designer had in mind. The question is not, "Would someone else familiar with standard industry practice have come up with this database design?", but rather, "Would someone else who was trying to accomplish the same valuable behavior (e.g., compactness) have come up with this database design?"

Once the plaintiff has met its initial burden by specifying both the abstraction and its intended valuable behavior, the court can properly frame the issues of proof and engage in factfinding on the standard elements of trade secret status: whether the matter was secret and whether it had any economic value. To continue with the hypothetical, is the database design a secret way to achieve compactness or is it something that would have resulted from known and accepted programming practices? If it is secret, does it add economic value, that is, is the program more valuable because of the behavior introduced by this design choice. It is, in any event, crucial that an abstraction alone does not indicate what constitutes a trade secret in software.

Finally, note that most of the design and organization in software is routine, i.e., based on common practice. As with any

other field in which there is a body of accumulated, routine practice, most software is written using well-known ideas, principles, and designs. This is particularly true for software intended for a commercial setting, because reliability is of paramount importance and the known (and well-tested) design principles are the most reliable. As a consequence, trade secrets in non-literal aspects of a program's design and organization are typically relatively few in number, e.g., most of the abstractions used to describe the program in the copyright context would not qualify.

B. Procedural Implications for Trade Secret Evaluations, Particularly in Cases Involving Both Copyright and Trade Secret Claims.

Figure 1 (following page 12) shows our proposed process for evaluating trade secret claims, depicted using a traditional computer science notation called a flowchart. We believe the proposed process fits well with the traditional legal mechanisms for evaluating claims before trial, such as judgment on the pleadings, summary judgment, and provisions for the liberal amendment of pleadings to accord with the changing nature of proof elicited during discovery, as well as with the liberal pleading standards of the Federal Rules of Civil Procedure.

In order to make possible a substantive technical evaluation of the claimed trade secrets, the plaintiff must specify to the court at step 2 (as a necessary element of his cause of action) the specific secrets and the value that the plaintiff claims that each of them produces in software at issue. These claims are

evaluated, possibly with the assistance of a court-appointed technical expert, at step 4, prior to permitting the plaintiff to carry out any discovery of the defendant's code. The rationale behind this is simple: the plaintiff ought to be able to specify what it believes are its own trade secrets independent of any review of the defendant's code; this will reduce the opportunity to manufacture or inappropriately enhance claims.

While we hope that it will be an exceptional circumstance, for completeness we include the possibility of augmenting the list of trade secret claims after discovery. As a result of our concerns that access to the defendant's source code may permit the plaintiff to distort the process, we suggest that such late-arising claims undergo especially careful scrutiny to ensure that they are technically appropriate and should properly be given to the jury for consideration.

In cases that involve allegations of both copyright infringement and misappropriation of trade secrets in computer software, we suggest that the trial court take up the trade secret claims first. This serves the interests of accuracy and judicial economy by permitting the case to move forward on the substantive evaluation of the plaintiff's trade secret claim while minimizing the disruption to defendant and defendant's business interests. It also limits the potential for prejudicial spill-over into the trade secret claim from the discovery of the defendant's code on the copyright claim.

The trade secret process can proceed as far as step 3--determining whether the plaintiff has technically valid trade

secrets in its code--before the beginning the development of the copyright issues in the case. At that point, whatever the outcome of step 3 of the trade secret process, technical evaluation of the copyright claims can begin. If no trade secret claims survive step 3, the copyright process begins alone; if some trade secrets do survive, processing of copyright claims can proceed in parallel. This makes sense because the next step in the trade secret process is discovery and, by and large, the same discovery materials will be relevant to both trade secret and copyright claims.

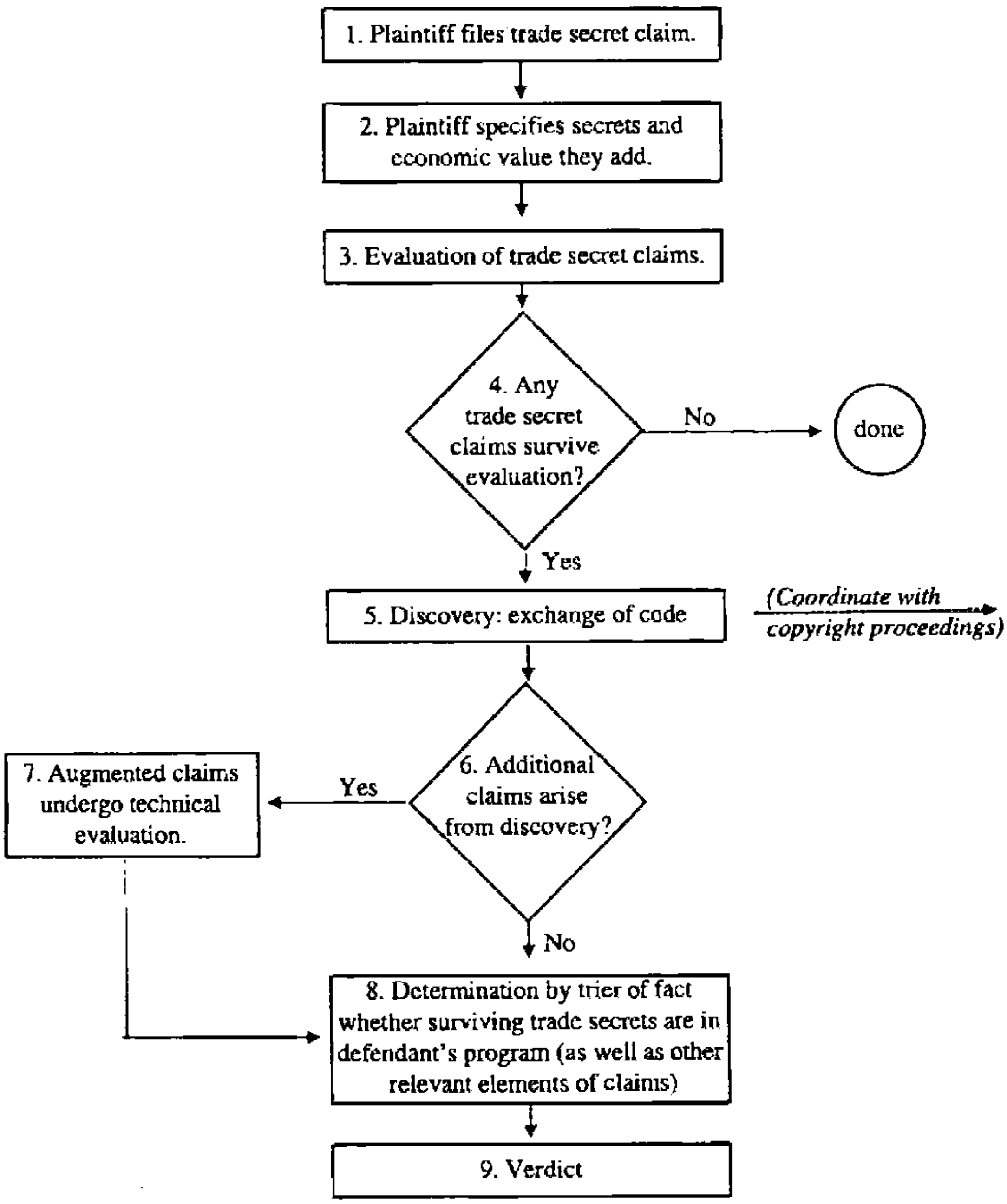


Figure 1: Proposed procedure for trade secret claim evaluation.

Notes on Figure 1.

Step 5: Exchange of code typically involves releasing each side's code, under non-disclosure, only to counsel and outside technical experts for the other side. Parties to the proceeding do not have access to the other side's code.

Step 8: A trade secret may be copied in a way that distributes the secret throughout code. In such circumstances, there may be no single place in the code that embodies the secret, and hence there are no abstractions that will embody the claimed secret. Instead, the secret must be sought by examination of the literal code, where it may be found in fragments throughout that code.

IV. PERFORMING THE AFC PROCESS

A. Meeting the Considerations of Technical Complexity: Refining the Magnitude of the Task.

1. Qualified Technical Experts Should Play an Important Role in the Abstraction Process.

Given the difficulty and complexity of creating a technically accurate set of abstractions, this Court should clarify that this is properly a task for a qualified expert in the field. As U.S. Court of Appeals Judge John Walker, author of the Altai decision, has noted, "Most juries, and most judges (myself included), are less than completely comfortable with the concepts and terminology of computer programs and need extensive education in order to make intelligent decisions." "Protectable 'Nuggets' Drawing the Line Between Idea and Expression in Computer Program Copyright Protection," 44 J. Copyright Soc'y U.S.A. 79, 92 (Winter 1996).

2. Refining the Task Will Reduce Technical Complexity.

In the interests of efficiency and manageability, the abstraction process should be carried out on a well-focused body of software. If the program is small, it may be practical to abstract the entire thing. When possible, this is advantageous because the abstraction and filtration processes can be carried out on the plaintiff's code prior to discovery of the defendant's code, yielding a clear picture, early in the process, of the exact contours of the protectable expression in the plaintiff's program.

Given the large size of most commercial programs, however, the effort involved in abstracting and filtering the entire program is enormous and would yield a daunting number of

exhibits. The court would likely be overwhelmed with paper and with the filtration effort of determining protectability for every element of the plaintiff's program, even if only a small part of that program is at issue in the claim of infringement. Thus, the obvious subset of code for analysis is that portion of the software that the plaintiff alleges the defendant copied.

3. The Plaintiff Must have Access to the Defendant's Code in Order to Focus the Inquiry.

In many instances, the initial evidence of copying is a relatively insignificant incident that turns out to be only the tip of the iceberg. For example, one case that eventually involved claims of extensive code theft (both literal and non-literal) as well as a variety of other serious charges began when a software developer noticed that a competing program built by former employees of his company displayed the same misbehavior that he knew to be present in his own code. As a piece of misbehavior, the behavior could not be motivated by the task, suggesting the possibility of copying. When this initial evidence of copying is sufficient to support a complaint, the plaintiff will need access to the defendant's code in order to determine the full extent--if any--of the problem.

We recognize the difficulty this access to code presents: if the plaintiff is permitted to examine the defendant's code before abstracting his own code, the plaintiff may "mine" the defendant's code for potential foci of alleged copying, thereby distorting the inquiry. But we believe some form of difficulty is unavoidable. If plaintiff instead must perform the abstraction and filtration before

seeing the defendant's code, it may feel that the only way to protect the claim is to abstract the entire program, resulting in an impractical--and wasteful--amount of work for both the plaintiff and the court. If, in the interests of economy, the plaintiff abstracts only those parts of its code that it suspects at the outset were copied by the defendant, the plaintiff's exhibits are likely to be incomplete and require augmentation after it has had the opportunity to examine the defendant's code, in which case the possibility of "mining" reappears. In any event, we believe that manufactured claims of copying become evident at the filtration stage, where the court's expert can raise such observations and the court can deal appropriately with them at that time.

B. A Process for Plaintiff's Performing the AFC Evaluation

Having performed a thorough examination of the defendant's code to evaluate the extent of copying, if any, the plaintiff's technical expert can then prepare a well-focused set of exhibits describing the relevant portions of the plaintiff's code at multiple levels of abstraction. This set of exhibits is the plaintiff's initial contribution to the AFC process. (See Appendix A for a hypothetical example).

At this point, the defendant must have access to the plaintiff's code in order to make an independent judgment as to whether the abstractions are accurate. This must include access to the plaintiff's entire program, even though the abstractions may deal with only a small part of the program, in order to test for errors of omission.

The court will now have before it a set of exhibits describing a carefully selected portion of the plaintiff's code, which the plaintiff claims is both protected and infringed. It is now in a position to evaluate the first of these claims independent of the second, and we suggest it do so, proceeding with the filtration test.

This approach has several significant advantages. If the exhibits contain no protectable material, the case is over, cf. Lotus Dev. Corp. v. Borland Int'l. Inc., 34 U.S.P.Q. 1014 (1st Cir. 1995), aff'd. by an equally divided court, 116 S.Ct. 804 (1996). (holding that AFC test unnecessary when menu command hierarchy constituted a "method of operation" that is uncopyrightable under 17 U.S.C. 102(b)). The defendant is spared unnecessary expense, yet the plaintiff has had a full opportunity to find copying in the defendant's program. As a matter of practicality, this may be a significant virtue in a world where litigation may be used as an economic weapon, particularly against smaller companies, of which there are many in the software world.

An additional advantage arises from the reduced workload presented to the court: by focusing the filtration purely on the plaintiff's exhibits, there is no need for the court to examine or display the defendant's code at this point. Finally, if the plaintiff understands that infringement claims will be required to pass the filtration test relatively early in the litigation process, independent of any court consideration of the defendant's code, the plaintiff may perform more careful analysis before filing infringement cases.

After the filtration is complete, the court should require the plaintiff to augment its exhibits with specific references to the defendant's code, indicating the exact lines of the defendant's code that it alleges are copied from the plaintiff's software[4]. Plaintiff's specific allegations will further focus the court's task and enable the defendant better to perform the next step, having the defendant's expert prepare abstraction exhibits of the defendant's code.

The inherent flexibility of the abstraction process, involving as it does many judgment calls, requires that the court permit each side to abstract its own code and to prepare its own abstraction exhibits embodying that analysis. The fact that the district court in the present case apparently permitted the plaintiff to prepare the abstraction exhibits for both parties, subject to defense objections, demonstrates once again a level of uncertainty as to the technical limitations of the AFC process.

The AFC process is not a magic wand that will eliminate disagreement; it is instead a framework within which the parties may carry out a discussion. As experts in the field, we anticipate that there will, in fact, be disagreement over the abstraction exhibits. Indeed, these arguments may hold the key to the remainder of the case. For this reason, the court should require the parties to offer technical grounds both in defense of their choices and when challenging the choices made by the opposing

---

[4] Even where the plaintiff alleges non-literal copying, the plaintiff should be able to cite specific lines of the defendant's program that embody the non-literal element(s) allegedly copied.

party. Even so, if the parties cannot reach agreement on the challenged exhibits, this Court should advise the district courts to consider seriously the appointment of a neutral expert of its own in order to assist in arriving at a consensus as to what constitutes an accurate, specific, and complete description of the code in question.

With these agreed-upon exhibits in place, the court is now ready to begin the comparison process. The trier of fact will now have before it a set of exhibits characterizing specific elements from plaintiff's program that are subject to copyright protection, along with corresponding elements from the defendant's program that the plaintiff alleges are infringing.

We summarize this set of suggested steps in the AFC process in Figure 2. Note that in step 2 we suggest some form of early technical evaluation of the plaintiff's initial copyright infringement claims; that is, the plaintiff should be required to establish at this early stage of litigation some minimal technical evidence of infringement in order to proceed with its claim. We believe that our approach is consistent with federal procedure for the evaluation of cases before trial, particularly in the form of motions for summary judgment. See, e.g., Fed.R.Civ.P. 56; cf. Fed.R.Civ.P. 11.

We further suggest that this Court encourage trial courts to have the plaintiff meet this minimal burden through the independent evaluation of the plaintiff's claim by a court-appointed expert or by support from disinterested third parties, concerning the technical substance of the initial evidence for infringement. Such a barrier,

even if relatively modest, would set some threshold level of technical substance to deter the filing of cases for little more than economic reasons (i.e., to interrupt the business of a competitor). We believe that, given the extensive time, effort, and expense of fighting a lawsuit, this process is a worthwhile way of minimizing the undesirable economic costs of unnecessary litigation.

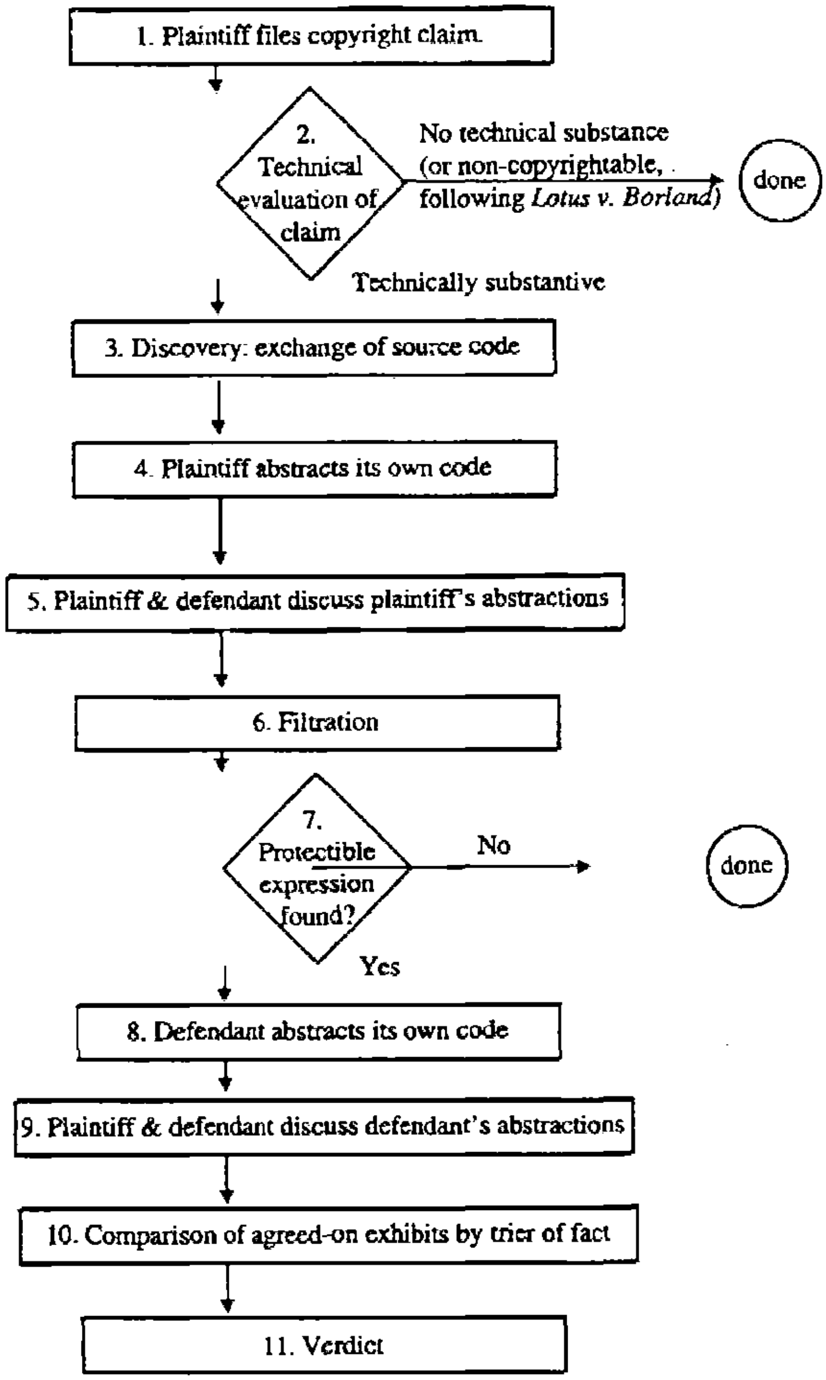


Figure 2: Proposed procedure for performing the AFC test.

C. Meeting the Considerations of Technical Complexity: Providing Standards for Abstraction

Although there is considerable art in abstracting a computer program, it is both possible (and useful) for this Court to give both the lower courts and litigants additional guidance regarding standards for abstracting software in cases involving allegations of copyright infringement. Such guidance is necessary because the process of abstracting programs is sufficiently unstructured that an interested party may bias the abstraction process. With this in mind, we have already suggested that each party prepare its own abstractions. In order further to reduce the opportunity for bias, we suggest here guidelines for the production of abstractions that may serve as a yardstick of technical quality: (1) we identify aspects of a program that generally ought to be subject to abstraction; (2) we call for clear reference to specific behavior and code being abstracted in each exhibit; (3) we require completeness in each abstraction exhibit at each level of detail; and (4) we suggest standardized graphic conventions for all exhibits.

1. Routine Abstractions Include Control Structure, Data Structure, Data Flow, Information Architecture and Textual Organization of the Code.

The program's control structure is the sequence of operations that it carries out, often indicated with a well-established graphical language of boxes and arrows called a flowchart (a format we used in Figures 1 and 2). Control is frequently the most complex aspect of a program; a complete set of control abstractions may

have many levels of detail. The data structures indicate the way in which individual elements of information are stored in the program; in the earlier checkbook hypothetical, for example, data structures are used to store the sorts of information found in a check register (*e.g.*, check number, date, payee, amount). The data flow is a description of how information flows through a program; that is, how the information for a check flows from the register where it is entered, to the check itself. The information architecture of a program indicates the overall organization of the data used by the program, often in the form of the organization of databases.

All of these abstractions concern the behavior of the program. Because programs can be viewed in terms of both their behavior and their text (treating the source code as a body of text), we can also describe the organization of the textual code itself at several levels of abstraction, ranging for example, from individual routines, to files containing multiple routines, to directories containing multiple files.

2. Abstractions Should be Specific and Precise.

As one indicator of such precision, there should be no ambiguity about what behavior is being described and what body of literal code is being abstracted. This enables evaluation of the accuracy and completeness of the abstractions. Courts should thus require: (a) that labels on abstraction exhibits clearly specify the program behavior at issue; and (b) that each component of an abstraction exhibit refer clearly either to more detailed exhibits or to literal code. In Appendix 1 we provide an example in which each component of every abstraction indicates where more detail can

be found, by reference to other exhibits containing other (lower level) abstractions, by naming specific routines in the code, or (at the lowest level) by citing specific lines of code.

3. Abstractions at Any Given Level Should Be Complete.

Such completeness will ensure that the exhibits present an entire and accurate picture of the program at any chosen level of detail. As one example, Figure 1.1 of the Appendix shows all three capabilities of the program at that level of abstraction. Subsequent figures provide additional levels of detail for only one of the components of that diagram (the box labeled "Balance checkbook"), but at the level of detail in Figure 1.1 the depiction is complete. The abstraction effort can be focused on the code relevant to the case at hand in the manner shown in the Appendix, i.e., by cutting off the abstraction process below a certain level for the irrelevant parts of the code.

4. Parties Should Adopt Explicit and Consistent Graphical Conventions for All exhibits.

The adoption of consistent graphical conventions will permit the trier of fact to make sensible comparisons of the exhibits. As we indicate in the Appendix, to a trained eye, some of the graphical organization of the abstraction exhibits is informative, while other elements are accidental (e.g., the left-to-right order of certain of the boxes). If the parties cannot agree on a set of conventions, the court's expert should assist the parties and the court in reaching a uniform set of standards for the exhibits.

V. CONCLUSION

The AFC process is a difficult and often complex procedure involving a large number of technical judgments regarding computer software in a rapidly evolving area of intellectual property law. For the foregoing reasons, regardless of the disposition of this case, we suggest that this Court take the opportunity to provide further guidance to the lower courts regarding the implementation of the AFC process, particularly in cases involving claims of both copyright infringement and trade secret misappropriation.

Dated: September 25, 1997

Respectfully submitted,

Marc M. Arkin

Marc M. Arkin

140 West 62nd Street

New York, New York 10023

(212) 636-6850

Of Counsel to Amici Curiae

Computer Scientists

Appendix I: Sample Abstraction

Program to handle an electronic version of a checkbook

In this Appendix we provide a description of a very simple program -- one designed to carry out a number of straightforward tasks involving a checkbook -- as a way of making concrete the notion of levels of abstraction of a program and as a way of illustrating some of the technical standards that courts may require when requesting abstraction exhibits.

The small size of the program makes it possible to consider abstracting the whole thing, yet even here we focus on just a segment of the entire body of software in order to keep the example of reasonable size.

Control Structure Abstractions

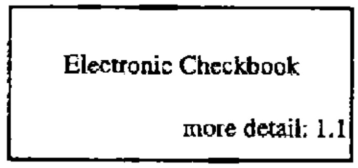


Figure 1: Program at the highest level of abstraction.

[The most abstract description of the program is its overall purpose or function.

Note that each abstraction makes reference either to another diagram that supplies the next most detailed view, or to a specific body of code that it abstracts.]

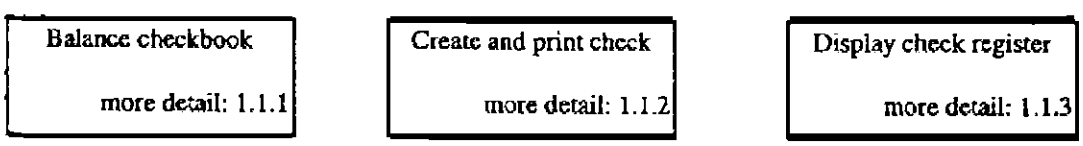


Figure 1.1: Next more detailed level of abstraction.

[Note that the left to right order is irrelevant; the diagram shows only that there are three separate capabilities in the program.]

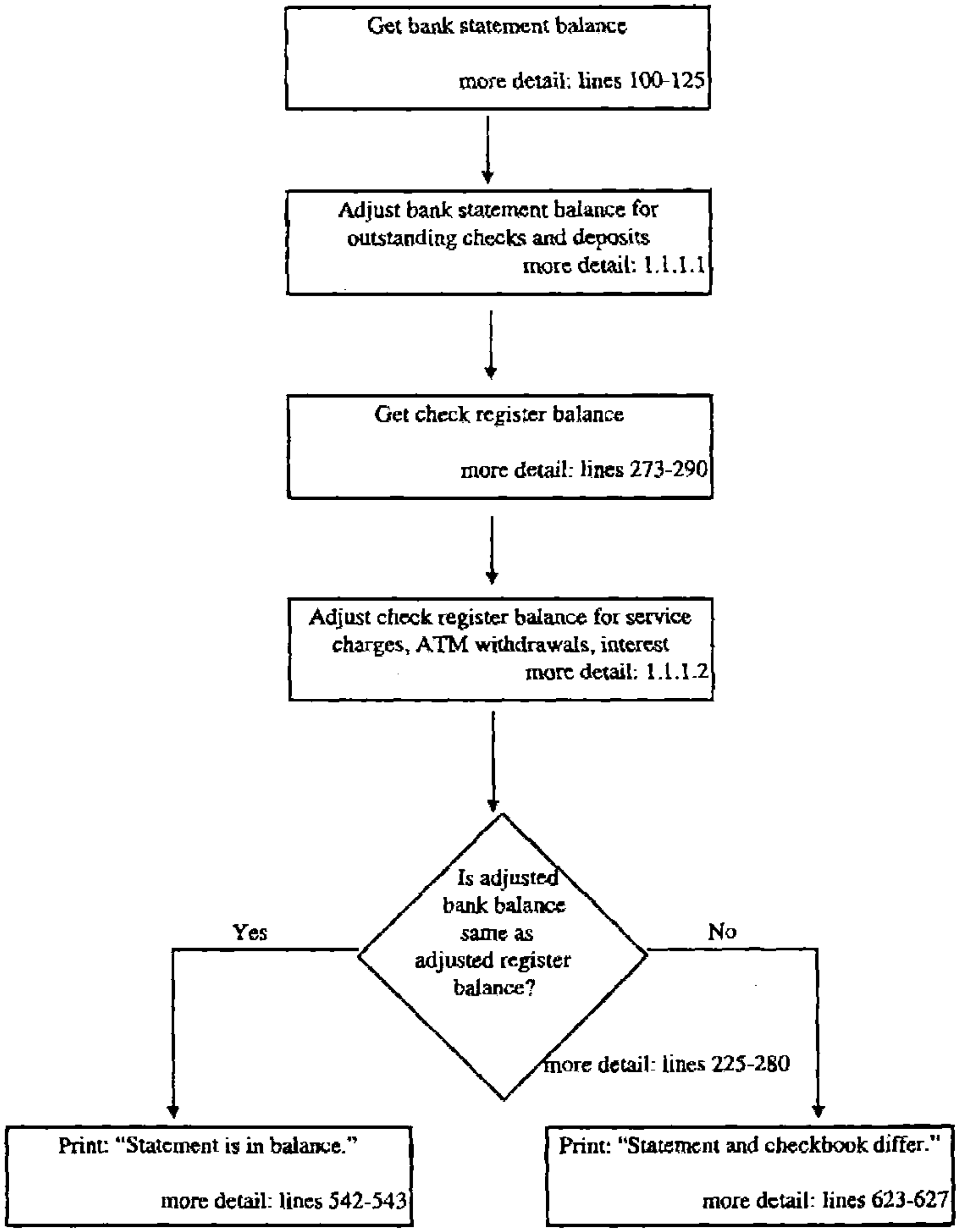


Figure 1.1.1: More detailed view of "Balance checkbook" from Figure 1.1.

Figure 1.1.1 continued

[Arrows indicate a specific sequence of actions. A technical expert would recognize that while the first two boxes had to be done in that sequence (i.e., the order is required by the task), and the third and fourth boxes had to be done in that sequence, the third and fourth boxes could be done before the first and second. That is, it doesn't matter whether we adjust the bank balance or the check register first.

. A diamond is the traditional flowchart symbol indicating a decision.

Some of the boxes here reference specific lines of code from the program, indicating the final level of abstraction, the literal code itself.

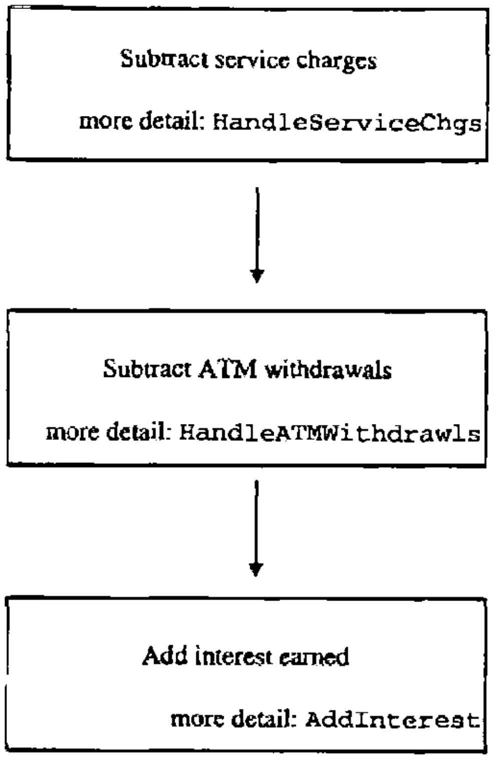


Figure 1.1.1.2: Next level of detail for "Adjust check register balance for service charges, ATM withdrawals, interest" in Figure 1.1.1.

[The boxes here indicate the name of a routine in the code that carries out the behavior described by the box. This is equally as specific as citing lines of code and is often more comprehensible.

A technical expert would recognize that the while the program being described ordered these particular steps in the sequence show, the order chosen is not at all constrained by the task and can be selected at will by the programmer.]

Data Structure Abstractions

[Words in ***Bold Italic*** font (other than headings) are the names of data structures or data abstractions in the program. As previously, the abstractions start with the most abstract and proceed to the most detailed.]

A ***Check Register*** contains one or more:

Check Register Entries, each of which is one of the following:

Check

Deposit

ATM Withdrawal

Interest

Service Charge

Note

[A technical expert would realize that the order in which this list is given is irrelevant and may be chosen by the programmer. Each of these entries should be further described, we use ***Check*** as an example.]

Each ***Check*** contains the following information:

Check Number

Date

Payee

Memo

Amount

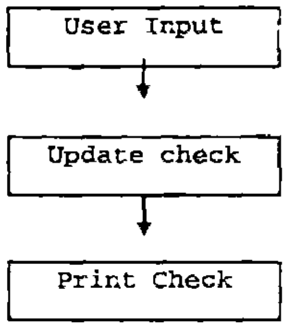
The specific data structure for a ***Check*** is:

NNNN DD MMM YYYY PPPPPPPPPPPPPPPPPPPP MEMOMEMO AAAAA.AA

indicating that there are four digits for the check number, two for the day, three characters for the month, four digits for the year, twenty characters for the payee, eight for the memo, and finally the amount, indicated using five digits before the decimal point and two after.

Data Flow

[As one example we show the flow of information about a check from the user's initial input to the printing of the check.]



Information Architecture

The program has two databases:

CheckBooks is a database containing information about each check book that the program is managing for us. Each database entry indicates the ***account number, owner***, and ***co-signer*** for the checkbook.

CheckRegisterData is a database containing information about transactions in each checkbook. Each database entry indicates a ***check, deposit, ATM withdrawal, interest***, or ***service charge***.

Physical Organization of Code

The directory ***CODE*** contains all of the source code files, which are:

UserInteraction

BalanceCheckBook

RegisterDisplay

WriteChecks

[A more detailed description would indicate the individual routines that make up each of these files.]

The directory ***DATA*** contains all of the system's data structures and databases.

[A technical expert would recognize that the physical division of the program into source code and data directories is a routine practice common in the field and done in part for reasons of efficiency.]

APPENDIX II: TERMINOLOGY

We have attempted to minimize the amount of jargon used above, but some technical terms are unavoidable. This appendix provides brief definitions of these terms sufficient to remove any mystery surrounding them.

Source code: The text of a program as written by the programmer; it generally looks like a combination of mathematics and English. Computer programs are written using specialized languages designed for this purpose; there are hundreds of such languages though perhaps only a dozen or so are in wide use. Commonly used languages include COBOL, BASIC, C, and (recently) JAVA.

Object code: Source code is translated from its English-like notation into a much more detailed language called object code, that is expressed as a collection of 1's and 0's. This language can be understood directly by the computer as instructions to carry out.

Control structure: one of the most basic things a programmer does is instruct the computer as to the sequence of events that should occur (i.e., what the program should do and when). There are a number of standard ways to control the sequence of events, these are called control structures.

Data structure: another standard task of a programmer is organizing the information that the program is to use. A data structure is a specification of the form and content for information stored in the program. A data structure for a check, for example, might indicate that the relevant information was check number, date, payee, and amount, and would specify the exact form of each piece of information.

Algorithm: an algorithm is a detailed specification of all of the steps necessary for carrying out a task. As an example, the monthly checking account statements from your bank often have the algorithm for balancing your checkbook printed on the reverse side of the statement. Note that an algorithm is a set of instructions to enable accomplishing a task, perhaps by a human; algorithms are not used only by computers. When done by a computer, algorithms use data structures and control structures. Algorithms can be described in flowcharts.

Comments: Virtually all computer languages make it possible to embed textual comments in the text of a program. Such comments are set off from the rest of the program by some textual cue, with the result that the computer ignores them. The comments are written in English and are intended to aid programmers in understanding the code.

File structure: the file structure of a program refers to the way it organizes information. Programs often need to read data from files and write information to files in order to keep permanent records. How these files are organized and used is an important part of the design of a program.

Because the source code of a program is itself a collection of files, we can also talk about the file structure of the source code itself (*e.g.*, how the text of the program is organized into component pieces, somewhat like the chapters in a book, or the volumes of a collection).

Data flow: A data flow diagram is a map of sorts, indicating how and where data are used throughout the program. It shows the "route" information travels as it is processed in the program.

EXHIBIT C

Sep 20, 2013

Artec Studio 9.2 is out! Introducing Auto-align tool and much more...

We are excited to announce the release of Artec Studio 9.2. This release constitutes a major leap forward for our software and includes features that we have been working on for many years.

So what exactly is new?

Auto-align tool

The Auto-align tool was years in the making. It allows you to make multiple scans and then align them AUTOMATICALLY with a press of a single button. That's it. You no longer have to align scans by hand, which could be a tedious and time consuming task.

Photogrammetry support

Photogrammetry is a widely used tool in industrial applications. Artec scanners and Artec Studio are now compatible with most photogrammetry solutions on the market. Using the two technologies together, in some cases, makes the scan data more reliable and the scanning process easier.

Artec is working with photogrammetry market leader, Aicon 3D Systems to make this bundle solution affordable and easy to use. Read more about Artec's cooperation with Aicon and photogrammetry.

New editing tools

New editing tools

We've added some interesting editing tools. They are designed to improve post-processing like the defeature brush which automatically erases imperfect surfaces. Edit faster, work better, create!

More options

- Ability to control how the flash behaves
- Individual settings for multiple users of a single software license
- ”Points and solid”- a smart render mode that displays your raw data in a way that is easy to work with.

There are many other minor changes and improvements. To see a full list, please refer to the release notes.

Try it out

Try all these great features for free. Download Artec Studio 9.2 trial version.

Related content

Artec Studio 16 nominated in TCT Software Awards 2022

Awards

Artec 3D launches the 2022 Artec Leo with twice the power and even greater precision

Artec Leo

New video: Artec Studio 16 — the power is in the software

Artec Studio

See all news

Sign up for Artec 3D's monthly newsletter

Useful guides, how-tos, and more

Email *

Sign up

☐ Artec 3D may send me offers and promotions

Products

Showcase

Services

Help

About

Luxembourg

20 rue des Peupliers
L-2328, Luxembourg

+352 2 786-1074

+352 2 600-8670

USA

2880 Lakeside Drive, #135
Santa Clara, CA 95054

+1 669-292-5611

China

Room 1410, 14/F
China Merchants Plaza (South), 333 Chengdu Bei Lu,
Jing'an District, Shanghai

+86 021 8036-0438

Showroom

2880 Lakeside Drive, #135
Santa Clara, CA 95054

+1 669-292-5614

EXHIBIT D

STRUCTURED LIGHT TECHNIQUES AND APPLICATIONS

1. INTRODUCTION

With modern advancements in computational methods, optics, and graphics computing, 3D scanning is rapidly becoming more prevalently adopted in society. High-density 3D scanning can be performed at rates of real time or faster, thus broadening the scope of applications to which these technologies can be applied. A structured light scanning system projects different light patterns, or structures, and captures the light as it falls onto the scene. It then uses the information about how the patterns appear after being distorted by the scene to eventually recover the 3D geometry. The potential speed of data acquisition, noncontact nature, the availability of necessary hardware, and the high precision of measurement offered by modern 3D structured light scanning technologies are what make them highly adoptable into industries such as medicine, biology, manufacturing, security, communications, remote environment reconstruction, and consumer electronics.

As the number of applications in which structured light techniques are employed increases, more interesting and challenging problems arise. It should be noted that there is not one 3D sensing technology that solves each issue and works as a general solution. The handbook by Zhang (1), which discusses many of the major 3D acquisition technologies, can be used to identify advantages and disadvantages of each approach. This article, however, focuses on discussing structured light scanning techniques.

Structured light techniques have benefited greatly from recent advancements in digital technology. To achieve real-time 3D data acquisition and reconstruction, much computational power is required, yet it can now be matched by today's modern computers; even some modern tablet computers can be used for these purposes. Given this, it is clear that the barrier to entry for some applications to use these technologies is quite low. However, in the past; if a manufacturing operation, for example, wanted to use these techniques, it may have been quite difficult to obtain the minimally required hardware, let alone to deploy the actual system itself. Modern advancements have ensured that the required hardware is relatively easily available. Software techniques required to reconstruct 3D data have improved greatly as well and these will also be discussed.

To be useful for a wide range of applications, a system that can capture and reconstruct 3D information in real time (online), instead of retroactively (offline), is necessary. Real time in this context typically refers to acquiring, processing, and visualizing the captured 3D data at speeds of at least 24 Hz (2). As mentioned earlier, advancements in hardware have made this a possibility, yet it is not a mere trivial task; many problems had to be solved before the technology reached this point.

As mentioned earlier, structured light scanning is only one method of 3D data acquisition. Within the area of structured light scanning, there are many different techniques, as well. One method stands out that has shown many advantages and derived from laser interferometers: digital fringe projection (DFP). The DFP technique involves varying sinusoidal patterns; using these methods, speeds of up to 120 Hz have been realized (3). However, the DFP approach has limits when it comes to its actual implementation, due to the sinusoidal pattern itself. Eight-bit grayscale images are required to display the sinusoidal patterns, yet modern projectors can only display 8-bit patterns at certain rates. Due to these speed limitations, among other reasons (such as nonlinearity calibration and correction), other methods have been developed. These other methods allow for increased speed while limiting the increase of system's complexity.

Such improvements to DFP include the squared binary defocusing technique (4). Whereas speed barriers were approached on today's digital image projection units with 8-bit images, 1-bit structured images can be projected much faster. By displaying 1-bit patterns through a defocused projector, the bits are blended together to reproduce a natural sinusoidal pattern. Another benefit, other than speed, of the binary defocusing technique is its lack of additional complexity due to calibration. As only two intensity values are being projected, nonlinearity calibration can be ignored. The smaller data transfer rate, increased rate of projection, and lack of complexity make this technique very advantageous and 3D shape measurement speeds greater than 120 Hz can be achieved. Using binary defocusing and digital light processing (DLP) platforms, Zhang et al. have been able to successfully develop a system that performed 3D shape measurement at tens of kHz (5).

This article aims to provide a review of the different principles used in structured light scanning technologies, 3D data acquisition, and, by extension, a summary of its many different fields of application. By no means does this article position itself as a comprehensive body of work covering all structured light technologies and their finer details; several other surveys have been written regarding structured light techniques that may be useful, as well (6,7). It should be noted that the content relies on published work, either in journals or in conference proceedings, done by us based on our own experiments or by others in the field.

Section 2 will cover different structured pattern encodings. Section 3 will detail the steps required to properly calibrate a structured light scanning system. Section 4 will discuss structured light scanning at real-time speeds including DLP technology, 3D data acquisition, processing, and visualization. Section 5 will outline several of the many fields in which 3D structured light scanning technologies are proving themselves to be useful, including communications, human-computer interaction (HCI), entertainment, medicine and biology, security, and remove environment reconstruction. Finally, Section 6 will conclude this article along with discussing future directions for 3D scanning and, specifically, structured light scanning.

2. STRUCTURED PATTERN ENCODINGS

A traditional way to obtain 3D depth is to mimic how stereo vision in the human eyes works using two different cameras (8). If a point or feature is contained within

J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering*.
DOI: 10.1002/047134608X.W8298

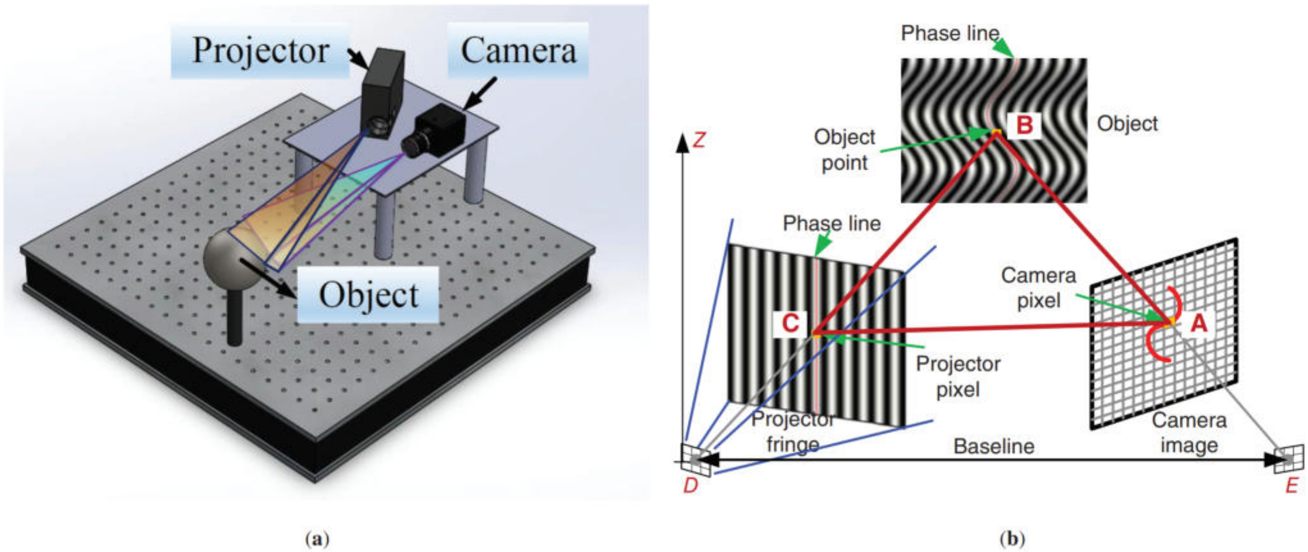


Figure 1. (a) Illustration of a structured light system containing one projector, one camera, and an object to be captured. (Reprinted with permission from Reference 12. Copyright 2014, Optical Society of America.) (b) Schematic diagram of a 3D structured light imaging system using fringe projection. (Reprinted with permission from Reference 3. Copyright 2010, Elsevier Limited.)

each camera image, and the physical relationship between the two cameras is known, then the point's depth can be recovered via triangulation. One of the issues with this approach is that it relies on the texture variation of the object being observed. If there is not a high texture variation in the object, then corresponding points, or feature points, are more difficult to find; this, in turn, negatively affects the stereo vision system's measurement accuracy. Methods such as spacetime stereo (9–11) have been developed to address the limitations of stereo vision systems, yet they themselves have drawbacks such as a limited field of view and a difficulty in reaching pixel-level resolutions (3).

Structured light systems work in a similar way to stereo vision systems; however, one camera is replaced with a projector. To overcome the issue of relying on the texture variation in the object being scanned, the projector can project its own special textures, or patterns, onto the object. These patterns, called structured encodings as they essentially encode the surface of the object being captured, can then be utilized within the frames captured by the system's camera. If the relationship between the two devices is known, calibration is discussed later in Section 3, then triangulation can be used, as before with stereo vision, to recover depth information. With this generalized structured light approach, measurement accuracy improves and rates of 3D data recovery are faster. An illustration of a physical structured light system along with its generalized principle is shown in Figure 1.

When using a structured light approach to obtain 3D data, the type of structured pattern projected onto the scene or object to be captured plays a key role in determining the speed, achievable resolution, and accuracy of the system (2). The following section provides a summary of several different structure pattern encodings.

2.1. 2D Pseudorandom Codifications

As mentioned earlier, the goal in encoding an object is to establish a one-to-one relationship between projector and camera pixels; this can be done by encoding the object with some pattern. One approach to encoding is to have a unique pattern in both the u and v directions, thus resulting in each pixel being uniquely represented. When implementing this method, a pseudorandom pattern can be generated or the natural speckle pattern of a laser source is used.

In the pseudorandom binary array approach, an array of size $n_1 \times n_2$ is encoded via a pseudorandom sequence such that any $k_1 \times k_2$ kernel within the array is unique. To encode the array, the polynomial modulo n^2 method is used, which can be mathematically described as

$$2^n - 1 = 2^{k_1 k_2} - 1 \tag{1}$$

$$n_1 = 2^{k_1} - 1 \tag{2}$$

$$n^2 = 2^n - 1/n_1 \tag{3}$$

The benefits of this method are that it is easy to understand and implement. Drawbacks of this method, however, include its sensitivity to noise and its inability to achieve a high spatial resolution. The resolution limitations are constrained by the resolution of the projector in both the u and v directions. If a structured light system is properly calibrated, however, a one-to-one mapping in both the u and v directions is not required; patterns could then vary in one direction and not the other. Figure 2 shows an example pseudorandom pattern generated via a *De Bruijn sequence*.

2.2. Binary Structured Codifications

Contrary to using pseudorandom codes to encode a scene, binary structured codifications encode a scene by using just two intensity values: all black (intensity 0) and all white

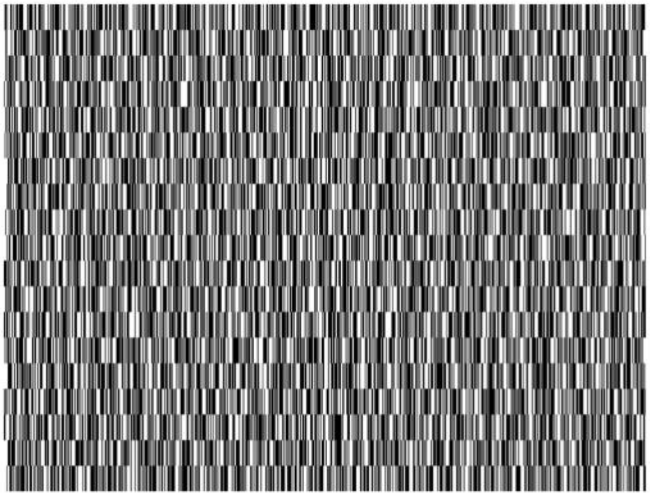


Figure 2. Example of a pseudorandom structured pattern generated via a De Bruijn sequence. (Image courtesy of Kenneth Joseph Paul from Purdue University.)

(intensity 1). Through the use of multiple, sequentially projected binary patterns, a scene can be uniquely encoded. That is, every pixel within a pattern can be uniquely represented with its own binary word (made up of 1s and 0s) through the sequential projection of different binary patterns onto a scene. Some sample binary patterns can be seen in Figure 3. This technique is advantageous due to its simplicity and robustness, especially in regard to noise and varying surface characteristics.

To actually acquire each pixel's unique codeword, the pixel's value (either 1 or 0) is determined for each pattern. After all of the patterns have been projected, the pixel's codeword within the camera's coordinate system will be known. With this, correspondence can now be established; this converts a pixel's codeword into the projector's coordinate system. If the system is calibrated, then the 3D coordinates can be recovered via triangulation between the camera coordinate, projector coordinate, and the scanned point.

Triangulation is achieved via

$$[u^c, v^c, 1]^T = [P_c][x^w, y^w, z^w, 1]^T \tag{4}$$

$$[u^p, v^p, 1]^T = [P_p][x^w, y^w, z^w, 1]^T \tag{5}$$

where P^c is the camera matrix, P^p is the projector matrix, (u^c, v^c) are the camera coordinates, and (u^p, v^p) are the projector coordinates. As can be seen in Figure 3, the codification only happens in either the u or v direction. The other direction will be unknown; this results in three equations and three unknowns. Solving these equations as a system of linear equations will result in the world coordinate (x^w, y^w, z^w). As mentioned earlier, it was assumed that this system was calibrated; more details on triangulation and how a structured light system may be calibrated will be discussed in Section 3.

The aforementioned benefits of the binary codification were its simplicity and robustness to noise, as well as surface variation. The disadvantages of this method are its limited spatial resolution and the number of patterns needed to uniquely encode a scene.

The spatial resolution using binary codes is limited by the projector's and the camera's resolution. Within one projected binary pattern, each pixel within a stripe will contain the same intensity value (either a 1 or a 0) as each stripe must be larger than one of the projector's pixels. Given this, however, the individual pixels within the stripe cannot be differentiated. As pixel-level correspondence between the camera and the projector cannot be obtained, a high measurement accuracy using the binary codification is difficult to achieve.

Another disadvantage of the binary codification is that many binary patterns are required to uniquely encode a scene. As only two intensity values are used to construct a unique codeword for each pixel, it takes n patterns to encode 2^n pixels. As the number of pixels increases, then so does the number of patterns required to achieve a dense 3D scan. The large number of patterns that need to be projected to obtain a single 3D frame may make this method not well suited for high-speed 3D scanning applications.

2.3. *N*-ary Codifications

To encode a scene with more than just two intensity values, as in binary codification methods where intensity values 0 and 255 are used exclusively, *n-ary codification methods* use any subset of values from 0 to 255; the largest subset of values would be to use the full range of values from 0 to 255. Figure 4 shows example n-ary patterns using different

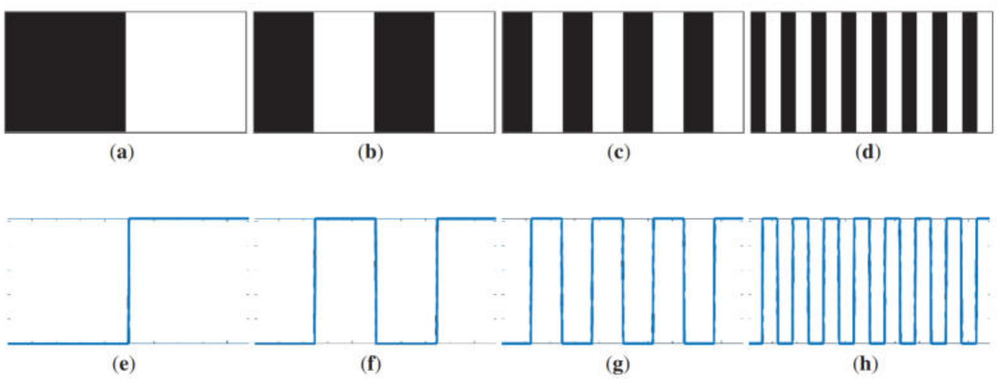


Figure 3. Sample codified binary patterns. To recover the codeword, a simple comparison of whether the pixel is black or white is checked for each image, and the result is placed within the corresponding bit of the codeword.

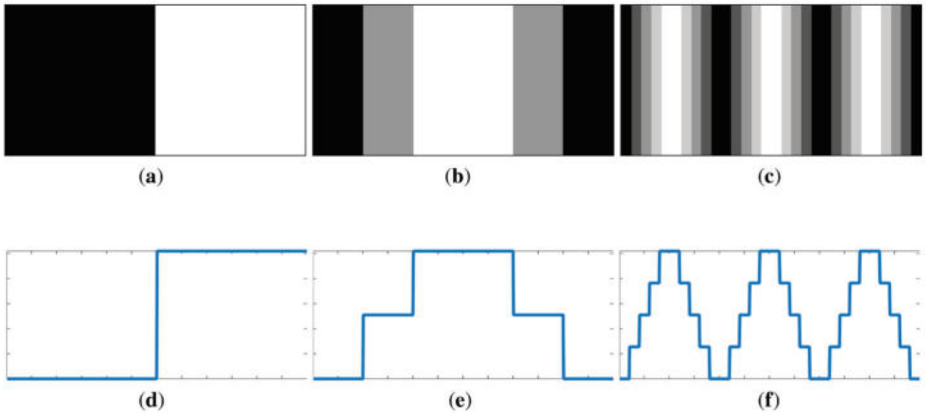


Figure 4. Sample n-ary binary patterns. The use of different intensity values other than simply 0 and 255 allows for the generation of more codewords in the same amount of images, or less, than binary patterns, for example.

intensity levels. A pixel's codeword can then be determined using the intensity ratio calculation (13,14). In general, this calculation involves the projection of two patterns onto the scene: one using the subset of intensities and one with a constant intensity across the image. The pixel's value then is the ratio of the pattern containing the intensities versus the constant pattern.

The use of many intensity values effectively decreases the number of patterns that need to be projected to encode a scene. One advantage of this fact is an increase in potential sensing speed. However, using many intensity values renders this method less robust to noise and surface variations. Also, as many values between 0 and 255 may be used, sometimes without much contrast between them, this method is very sensitive to the focus level of the projector and camera. In an attempt to avoid such issues, phase-shifting techniques can be used and different types of structured patterns are used.

2.4. Triangular Phase Codifications

One similar method for encoding a scene is to use *triangular* patterns (15,16). A triangular pattern refers to the shape a plot of the cross section of the pattern makes. One disadvantage of using a single pattern is its high sensitivity to noise. To address this, more patterns can be used to uniquely encode a scene.

To generate more triangular patterns, the original triangular structure can be *shifted* by some amount. For instance, two or more steps can be used to uniquely encode a scene (17,18). Figure 5 shows an example of one such triangular pattern described by Jia et al. and its cross section. An intensity ratio calculation can be used to recover the original triangular shape between the two patterns. The triangular shapes can then be removed to leave just the wrapped intensity ratio distribution. After being unwrapped, this can be used with an intensity ratio to height conversion to recover 3D geometry. Details about wrapped and unwrapped phases will be covered in the

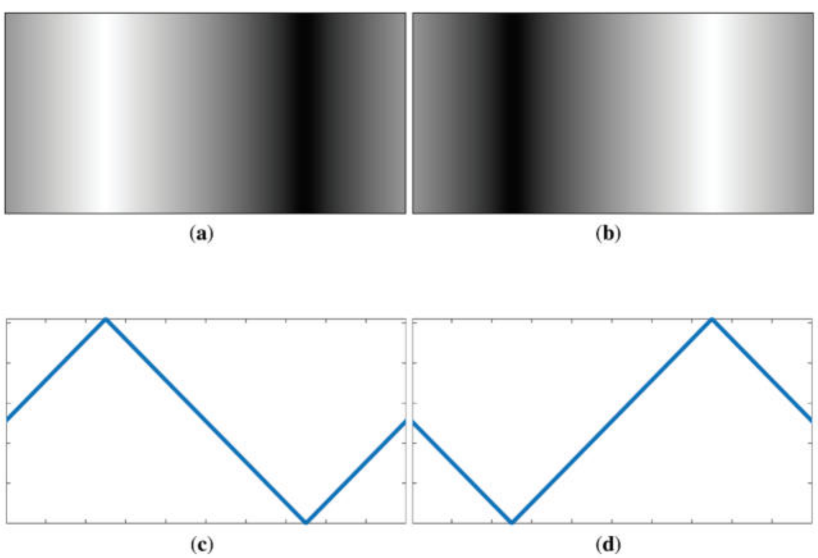


Figure 5. Example of a triangular pattern and its cross section.

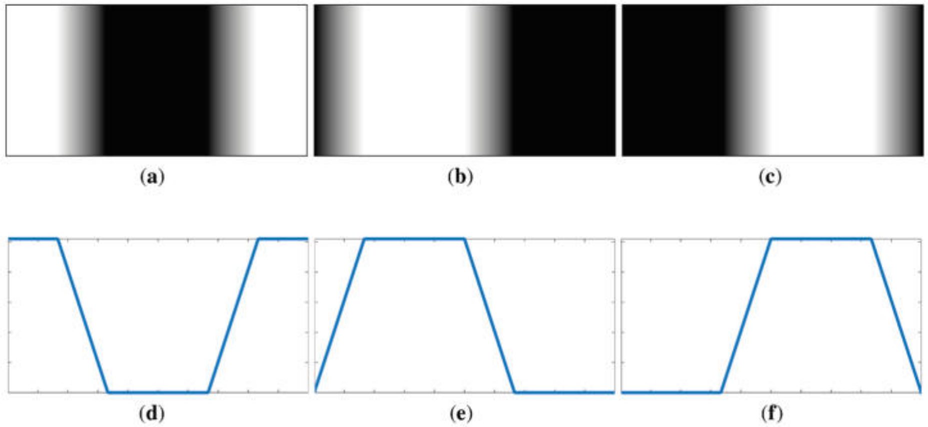


Figure 6. Example of trapezoidal patterns and their respective cross sections.

"*N*-Step Phase-Shifting Technique" section. The advantage of using triangular patterns is their relative simplicity and speed; however, these methods are sensitive to projector gamma nonlinearity and image defocus (17).

2.5. Trapezoidal Phase Codifications

To alleviate the drawback of sensitivity to image defocus by triangular structure approaches, Huang et al. (19) described a unique way to encode a scene using shifted patterns that represent trapezoidal structures. Figure 6 shows example trapezoidal structures and their cross sections as described by Huang et al. (19). This method maintains the speed offered by triangular approaches via the use of intensity ratio calculations, yet is six times less sensitive to noise (19). From here, the techniques to recover 3D data are similar to those used when using triangular structures: recover an intensity ratio map and remove its triangular shapes. The remaining intensity ratio map can then be used to eventually recover 3D geometry (20,21). Although this method boasts a low sensitivity to noise, it may still be slightly sensitive to image defocus. To address the issue of image defocus, sinusoidal phase-shifted patterns can be used; these will be discussed next.

2.6. Continuous Sinusoidal Phase Codifications

Although there are many types of structured patterns (e.g., binary, n-ary, triangular, and trapezoidal), many will appear to be sinusoidal in nature if they are properly blurred (3). With sinusoidal patterns, pixel-level resolution is possible as intensities vary across the image from point to point at known frequencies and therefore can be differentiated. Given this, the correspondence between the camera and the projector can be determined more precisely, thus increasing the potential sensing accuracy of such method.

These sinusoidal patterns, also known as fringe patterns, have been studied extensively in optical metrology due to their ability to reach pixel-level spatial resolution. To generate such fringe patterns, there are several methods. One of which is via laser interference; however, lasers inherently tend to have some speckle noise; this could degrade the potential accuracy of such method. An alternative method is the digital fringe projection, which uses a digital video projector to project computer-generated sinusoidal patterns onto the scene to be scanned. Instead of using intensity values to establish correspondence, phase information is used. One benefit of this is an inherent robustness to surface texture variation.

This phase information can be obtained via the Fourier transform profilometry (FTP) method (22), which only requires a single fringe pattern. Such a method may be sensitive to noise and surface texture variations, however. To alleviate these drawbacks and to achieve a scan of complex 3D structures, three or more sinusoidal patterns must be used (23) with their phase information shifted; examples of such shifted patterns are shown in Figure 7. This allows each pixel's phase value to be computed independently from any other pixel, hence why such a method is robust to surface texture variations as well as geometric variations. Due to the shift in the phase information between the three or more patterns, these techniques are called *phase-shifting methods* and will be discussed next.

N-Step Phase-Shifting Technique. Phase-shifting methods are popular within optical metrology for their multiple advantages (24):

- *Less sensitive to surface reflectivity variations.* As will be discussed later, phase-shifting methods typically compute a point's phase value via the arctangent function, and as surface reflectivity is constant for each pixel, it will be cancelled out.
- *Immune to ambient light influences.* The phase-shifting method uses and analyzes phase information instead of intensity information.
- *Dense 3D shape measurement.* Pixel-level spatial resolution is possible as each camera pixel contains its own 3D information, thus allowing for dense scans.
- *Can achieve high measurement accuracy.* Subpixel correspondence between the projector and the camera is possible if calibration is properly performed, thus allowing for highly accurate 3D scans.
- *Permits high-speed 3D shape measurement.* A scene to be scanned only has to be projected upon and processed once for a single scan; multiple passes are

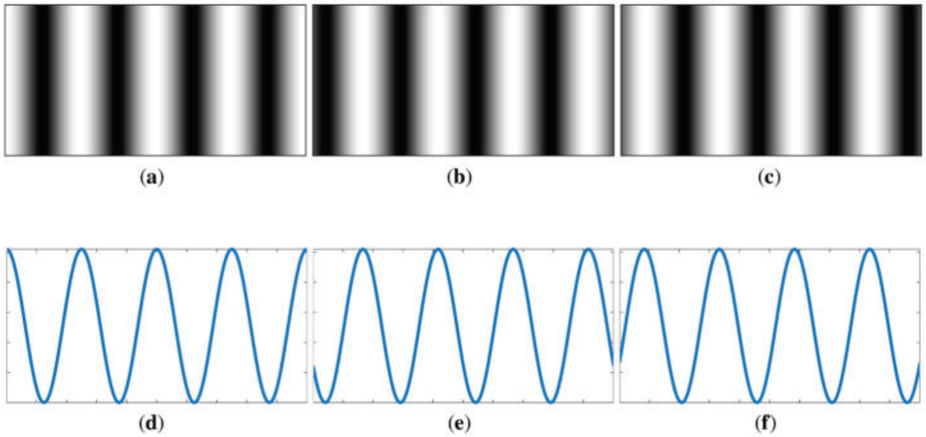


Figure 7. Sample sinusoidal patterns with phase shifts. These patterns can be computer-generated and sequentially projected upon a scene with a digital video projector.

not required; thus, fast measurement speeds can be achieved.

As mentioned earlier, three or more fringe images must be used if robust and accurate measurements are desired. There have been many different phase-shifting algorithms developed, however, including three-step, four-step, and the generalized N-step algorithms. An N-step phase-shifting algorithm with equal phase shifts is given as

$$I_n(x, y) = I'(x, y) + I''(x, y)\cos(\phi + 2n\pi/N) \tag{6}$$

where $n = 1, 2, \ldots, N$, $I'(x, y)$ is the average intensity given by

$$I'(x, y) = \frac{\sum_{n=1}^{N} I_n}{N} \tag{7}$$

$I''(x, y)$ is the intensity modulation given by

$$I''(x, y) = \frac{\sqrt{\left(\sum_{n=1}^{N} I_n \cos(2n\pi/N)\right)^2 + \left(\sum_{n=1}^{N} I_n \sin(2n\pi/N)\right)^2}}{N} \tag{8}$$

and $\phi(x, y)$ is the phase to be solved for via

$$\phi(x, y) = -\tan^{-1}\left[\frac{\sum_{n=1}^{N} I_n(x, y)\sin(2n\pi/N)}{\sum_{n=1}^{N} I_n(x, y)\cos(2n\pi/N)}\right] \tag{9}$$

Also obtainable from these equations is a 2D texture image without fringe stripes. This can be computed via

$$I_t = I'(x, y) + I''(x, y) \tag{10}$$

The "wrapped phase" ϕ obtained from equation 9 has a range of $[-\pi, +\pi]$ and thus contains 2π phase jumps. To obtain a continuous phase map, $\Phi^r(x, y)$, from the wrapped phase, a spatial phase unwrapping algorithm (25) is used. Such an algorithm locates the 2π jumps by comparing the phase values of the neighboring pixels and either adds or subtracts integer values of 2π.

Many spatial phase unwrapping algorithms have been developed and some are indeed quite robust. They all, however, suffer from the same set of drawbacks. First, they only work well for smooth or continuous surfaces without a high variation from one point to the next. Also, the unwrapped, *relative* phase, $\Phi^r(x, y)$, always refers to one point on the wrapped phase. It is difficult to uniquely correlate this relative phase value with the geometry's depth z value. To do this, absolute phase values are instead needed, which will be discussed next.

Absolute Phase Recovery. To obtain absolute phase pixel by pixel, more images are typically required, and a temporal phase unwrapping algorithm is usually adopted. Instead of comparing a pixel's value to that of its neighbor, a temporal phase unwrapping algorithm uses information from other phase values at the same camera pixel.

The 2π phase jumps within the wrapped phase occur when the projected fringe pattern contains more than one fringe stripe. That is, if the projected fringe was wide enough to cover the scene in a single stripe, the phase would be continuous and would not need to be unwrapped. One technique then is to use such a continuous phase map to unwrap an original wrapped phase map. Unfortunately, due to noise and hardware constraints, it is usually not possible to use a wide fringe pattern, with a single stripe that covers the entire measurement area, directly. Instead, two high-frequency fringe patterns may be used to generate an equivalent low-frequency fringe pattern.

Originating in physical optics, this *multifrequency phase-shifting* method's relationship between the absolute phase, Φ, the wavelength of light, λ, and the height, $h(x, y)$, is described as

$$\Phi = [C \times h(x, y)/\lambda] \times 2\pi \tag{11}$$

where C is a system constant. Now, for Φ_1 and Φ_2 with their respective wavelengths being such that $\lambda_1 < \lambda_2$, their difference is

$$\Delta\Phi_{12} = \Phi_1 - \Phi_2 = [C \times h(x, y)/\lambda_{12}^{eq}] \times 2\pi \tag{12}$$

The equivalent wavelength between λ_1 and λ_2 is

$$\lambda_{12}^{eq} = \lambda_1\lambda_2/|\lambda_2 - \lambda_1| \tag{13}$$

Now, if $\lambda_2 \varepsilon(\lambda_1, 2\lambda_1)$, we have $\lambda_{12}^{eq} > \lambda_2$. In reality, however, the only information available is ϕ_1 and ϕ_2, yet the relationship between absolute phase(s) and wrapped

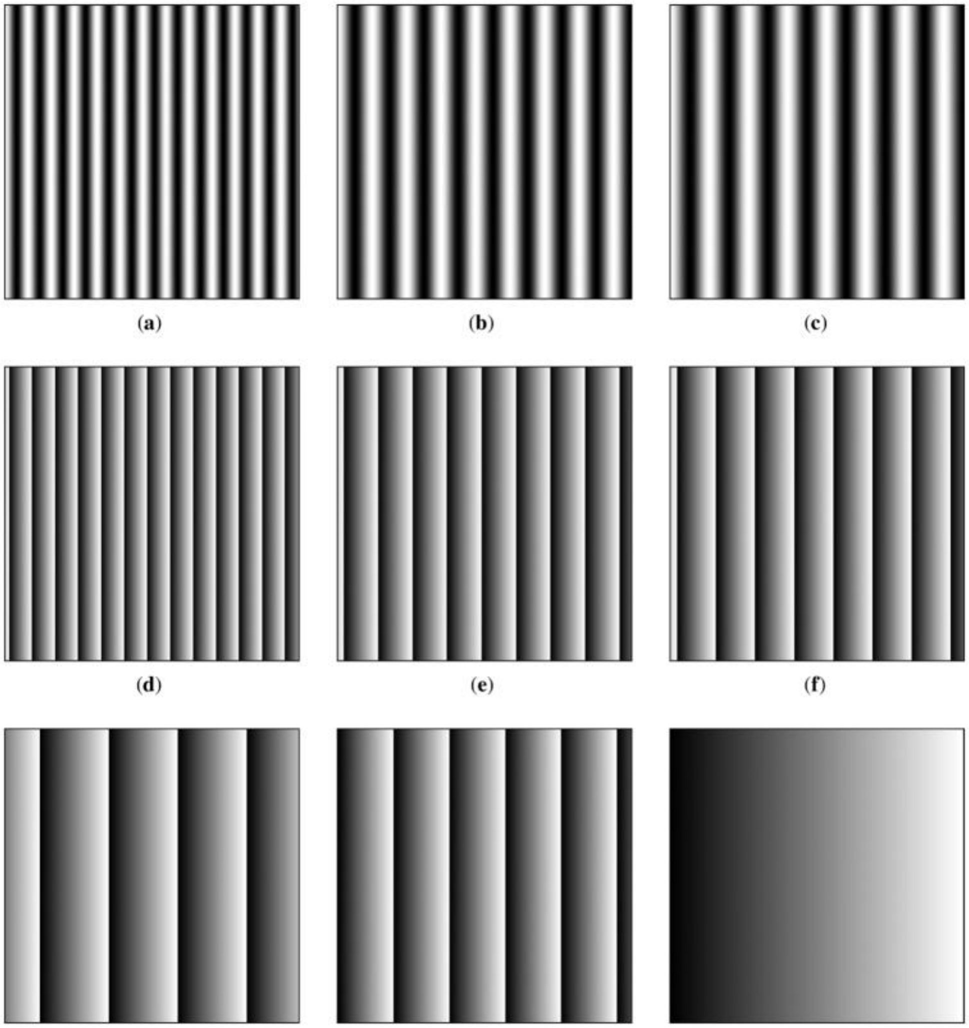


Figure 8. Multifrequency phase-shifting method. (a) One fringe pattern (λ_1 = 60 pixels). (b) One fringe pattern (λ_2 = 90 pixels). (c) One fringe pattern (λ_3 = 102 pixels). (d) Wrapped phase ϕ_1. (e) Wrapped phase ϕ_2. (f) Wrapped phase ϕ_3. (g) Equivalent phase difference $\Delta\phi_{12}$. (h) Equivalent phase difference $\Delta\phi_{13}$. (i) Resultant phase $\Delta\phi_{123}$ that can be used to eventually unwrap ϕ_1.

phase(s) can be described simply as $\phi = \Phi \pmod{2\pi}$ with 2π discontinuities.

The modulus operation converts the phase to a range of $[0, 2\pi)$. When the modulus operation is applied to equation 12, it similarly yields

$$\Delta\phi_{12} = [\Phi_1 - \Phi_2] \pmod{2\pi} = [\phi_1 - \phi_2] \pmod{2\pi} \tag{14}$$

The aim of the equivalent wavelength, $\lambda_{12}^{\mathrm{eq}}$, is to span the whole range of the image (i.e., $|C \times h(x, y)/\lambda_{12}^{\mathrm{eq}}| < 1$); doing this requires properly choosing the wavelengths for λ_1 and λ_2. If done correctly, the modulus operation does not affect anything and no phase unwrapping is required. This two-frequency method is hindered by noise, however. To achieve point-by-point absolute phase measurement, at least three frequencies of fringe patterns are typically used.

Given this, assume another set of fringe patterns is used with wavelength λ_3 in addition to the frequencies λ_1 and λ_2, which were previously used in the two-frequency method. Following the previous equations then, the equivalent wavelength between λ_1 and λ_3 will be $\lambda_{13}^{\mathrm{eq}} = \lambda_1\lambda_3/|\lambda_3 - \lambda_1|$. Then continuing on

$$\Delta\phi_{13} = [\phi_1 - \phi_3] \pmod{2\pi} = \{[C \times h(x, y)/\lambda_{13}^{\mathrm{eq}}] \times 2\pi\} \pmod{2\pi} \tag{15}$$

$$\Delta\phi_{123} = (\Delta\phi_{13} - \Delta\phi_{12}) \pmod{2\pi} = \{[C \times h(x, y)/\lambda_{123}^{\mathrm{eq}}] \times 2\pi\} \pmod{2\pi} \tag{16}$$

where $\lambda_{123}^{\mathrm{eq}} = \lambda_{12}^{\mathrm{eq}}\lambda_{13}^{\mathrm{eq}}/|\lambda_{13}^{\mathrm{eq}} - \lambda_{12}^{\mathrm{eq}}|$. As mentioned earlier, frequencies must be chosen so that $|C \times h(x, y)/\lambda_{123}^{\mathrm{eq}}| < 1$ so that the absolute phase can be obtained without spatial phase unwrapping. Using the absolute phase of the longest equivalent wavelength, the other wavelengths can be unwrapped (in reverse). The phase of the shortest wavelength is usually the one that is used to eventually recover 3D information as its measurement accuracy is approximately inversely proportional to the wavelength. Figure 8 provides a visual demonstration of the multifrequency phase-shifting method.

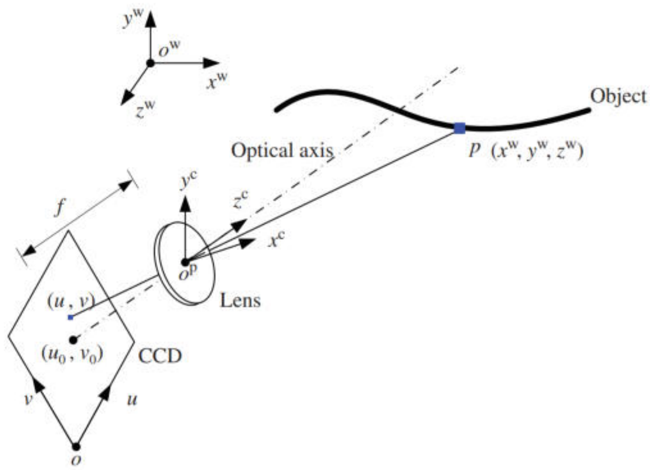


Figure 9. The pinhole camera model. The camera model is used to describe that an arbitrary point p in 3D space, with its own coordinate system, is transformed to the camera lens coordinate system, and finally, the 3D coordinates in the lens coordinate system are projected onto a 2D imaging space. (Reprinted with permission from Reference 40. Copyright 2006, International Society for Optics and Photonics.)

Alternatively, the temporal phase unwrapping can be realized by combining phase shifting with coding to uniquely determine the order of 2π, or fringe order for each 2π jumps. The fringe order could be encoded by a sequence of binary patterns (9, 26–30), a sequence of phase-encoded patterns (31–33), a single statistical pattern (34), or even embedding the statistical pattern into the sinusoidal pattern by adding a secondary camera using trigonometric methods (35–38).

3. STRUCTURED LIGHT SYSTEM CALIBRATION

Several times during the discussion of structured light techniques, it was assumed that the system was properly calibrated so that 3D information could be triangulated. Simply put, if the points on the camera can be matched with points on the projector, this information can be used to triangulate 3D points by solving linear equations. Then the goal of structured light system calibration is to determine the intrinsic and extrinsic matrices for the camera and projector so that their points may be matched.

The calibration of a structured light system involves the calibration of both the projector and the camera. Camera calibration has been extensively studied with the flat checkerboard method being a favorite due to its simplicity and speed. In Zhang's method (39), the camera is treated as the pinhole camera model. The checkerboard method then determines the camera's intrinsic parameters (e.g., focal length and principal point) as well as its extrinsic parameters (how to transform between the real-world coordinate system (x^w, y^w, z^w) and the camera coordinate system).

Shown in Figure 9 is the pinhole camera model used in Zhang's method with its intrinsic parameters being described as

$$A = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{17}$$

where (u_0, v_0) are the coordinates of the principal point (the intersection between the optical axis and the image sensor plane), α and β are the focal lengths along the u and v axes of the image plane, respectively, and γ is the property that represents the skewness of the u and v coordinates (in modern cameras, this is typically zero as the u and v directions are perpendicular to each other).

The extrinsic parameters are described as

$$[R, t] = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \tag{18}$$

where R is the 3×3 rotation matrix and t is the 3×1 translation vector.

An arbitrary point p, shown in Figure 9, has a world coordinate of (x^w, y^w, z^w) within the world coordinate system of $(0^w; x^w, y^w, z^w)$ and a camera coordinate of (x^c, y^c, z^c) within the camera coordinate system of $(0^c; x^c, y^c, z^c)$. The point's projection onto a uv image plane can then be described as

$$sI = A[R, t]X^w \tag{19}$$

where $I = [u, v, 1]^T$ is the homogeneous coordinate of the image point in the image plane, $X^w = [x^w, y^w, z^w, 1]^T$ is the homogeneous world coordinate for that point in the world coordinate system, and s is a scaling factor. It should be noted that equation 19 describes a linear camera model.

As mentioned earlier, a structured light system is different from a stereo vision system in that one of the cameras is replaced with a projector. Given that a projector cannot capture images like a camera, the calibration of a projector used to be quite difficult. Methods to calibrate the projector and the system, in general, have been developed over the years, yet they are often quite slow and lack high accuracy.

Some structured light system calibration methods first calibrate the camera and then use this to calibrate the projector (41); this, however, propagates any error of the camera calibration into the projector calibration. Ideally, the camera and projector should be calibrated independently. Zhang and Huang (40) developed a method that does this independent, and simultaneous, calibration of the camera and projector. Using both horizontal and vertical fringe patterns, a one-to-one mapping between camera points and projector points is established. Given this, the intensity map of the camera can be used to virtually generate images for the projector, thus permitting the projector to "capture" images similarly to a regular camera. Turning a projector into a camera, theoretically, means that stereo system calibration methods can again be used; these methods have been well developed and established. Also, as the two devices are calibrated independently, error from one device does not affect the other. Furthermore, as the devices are independent, they can be calibrated simultaneously, which results in a more rapid calibration procedure.

The (x, y, z) coordinates can be calculated using the absolute phase as a constraint along with the calibration information. Equation 19 for a structured light system's camera is then

$$s^c I^c = A^c[R^c, t^c]X^w \tag{20}$$

where s^c is the camera's scaling factor, I^c are the homogeneous camera image coordinates, A^c are the intrinsic parameters of the camera, and $[R^c, t^c]$ is the extrinsic parameter matrix for the camera.

Similarly then, the projection from the world coordinate system onto the projector's image plane is described as

$$s^p I^p = A^p[R^p, t^p]X^w \tag{21}$$

where s^p is the projector's scaling factor, I^p are the homogeneous projector image coordinates, A^p are the intrinsic parameters of the projector, and $[R^p, t^p]$ is the extrinsic parameter matrix for the projector.

Now there are six equations and seven unknowns: (x^w, y^w, z^w), s^c, s^p, u^p, and v^p. To solve for the world coordinates (x^w, y^w, z^w), one more equation is necessary and is provided by the absolute phase information. This is due to the fact that each point on the camera corresponds to one line with the same absolute phase on the projected image plane (40). If it is assumed that the fringe stripe follows the v direction, the relationship between the captured and projected fringe images can be described as

$$\phi^a(u^c, v^c) = \phi^a(u^p) \tag{22}$$

This equation provides a constraint for the phase, namely, each captured phase value corresponds to one fixed line on the projection image. By combining this equation with those two sets of projection equations for the projector and camera lenses, world coordinates (x^w, y^w, z^w) can be uniquely derived pixel by pixel (42). More detailed derivation can be found in Reference 43, and better and more accurate calibration approaches can be found in our recent papers (44,45).

Figure 10 portrays an example of 3D scanning with a properly calibrated structured light system using a DFP technique; specifically, a multifrequency phase-shifting method is used. Figure 10a–c shows a captured fringe pattern for each of the fringe frequencies. Figure 10d–f shows each respective frequency's wrapped phase. Figure 10g displays the equivalent wrapped phase λ_{12}^{eq}, and Figure 10h displays the equivalent phase λ_{123}^{eq}. From here, Figure 10d can be unwrapped and, using the calibration parameters, world coordinates can be derived. Figure 10i–l shows show different visualizations of the recovered 3D geometry.

4. REAL-TIME STRUCTURED LIGHT SCANNING

For a structured light scanning system to reach real-time speeds, several different approaches can be used. One of which is to use a single structured pattern (e.g., random codification or sinusoidal codifications) that spans the entire scene to be scanned. These methods are straightforward and fast, as only one image is projected, yet they are sensitive to surface texture variations and have a limited spatial resolution. Alternatively, several structured patterns may be used, as discussed in the previous sections, with the patterns being projected, captured, and processed rapidly.

Many methods have been introduced that involve projection of several different patterns from stripe boundary binary codes (46,47) to using color-encoded structure patterns (48–52). The stripe boundary code method, which projects a sequence of binary level structured patterns, suffers from spatial resolution limitations due to being constrained by the resolution of the projector. Using color-encoded structure patterns allows for higher speeds, as a structured pattern can be encoded with three-color structured patterns, yet this method's accuracy is affected by the surface color of the captured object; some color signals are lost depending on what colors are encoded and if they are similar to the color of the object. This is one example why black and white (monochromatic) structured patterns are typically used, instead.

Using monochromatic, sinusoidally varying structured patterns, Zhang and Huang (40) developed a 3D scanning system that, via the use of a fast three-step phase-shifting algorithm, could capture, reconstruct, and display 3D geometry in real time. To make this possible, the projection mechanisms employed by single-chip DLP were used. DLP projectors automatically and very rapidly project through their RGB channels. Instead of projecting each color channel in one image, three separate images, the three sinusoidal structures, can be projected. Zhang and Yau (53) were able to achieve 3D surface measurement speeds of 60 Hz with more than 300K points per frame. The technology used to realize this method will be discussed in this section.

4.1. Digital Light Processing Fundamentals

Digital light processing technologies were first conceived at Texas Instruments in the 1980s. Every DLP projection unit, at its core, has an extremely precise light switch called the digital micromirror device (DMD). The DMD chip is made up of microscopic mirrors with each mirror corresponding to one pixel of light in the eventual projected image. Each micromirror is set on its own hinge, along an array of other micromirrors, within the chip. Each mirror is controlled by electrostatic forces that turn a mirror either on, $+\theta_L$, or off, $-\theta_L$. If a mirror is switched on for 0% of a projection frame, no light for that pixel will be emitted (i.e., it will be black). Conversely, if a mirror is on 100% of the time, then the pixel will be white. By flipping this mirror on and off at different rates within a projection frame, different grayscale intensities can be realized for each micromirror. The principle of the micromirror can be seen in Figure 11. To achieve color projections, a color wheel is used. This color wheel, containing red, green, and blue filters, spins at very fast rates and thus sequentially projects a red channel, a green channel, and a blue channel. Through this method, the resulting projection as viewed by humans is a single color, not three individual colors, due to the speed of rotation of the color wheel. To produce a grayscale value, a DLP projector uses time integration (54).

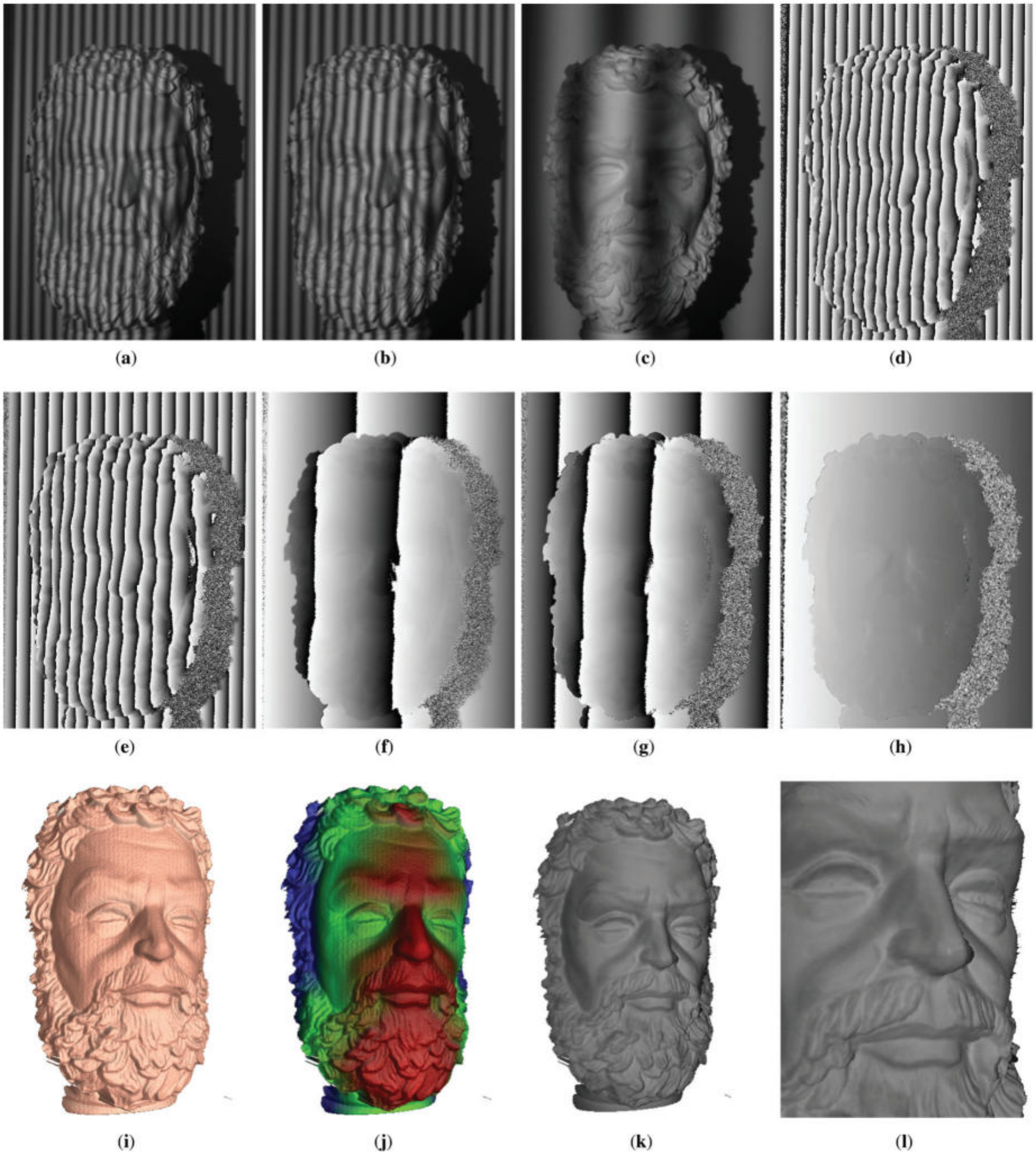


Figure 10. Example of 3D frame capture using a multiwavelength phase-shifting method with a properly calibrated structured light system. (a) One fringe pattern (λ_1 = 30 pixels). (b) One fringe pattern (λ_2 = 36 pixels). (c) One fringe pattern (λ_3 = 231 pixels). (d) Wrapped phase ϕ_1. (e) Wrapped phase ϕ_2. (f) Wrapped phase ϕ_3. (g) Equivalent phase difference $\Delta\phi_{12}$. (h) Equivalent phase difference $\Delta\phi_{123}$ that can be used to unwrap ϕ_1. (i) Reconstructed 3D data from the unwrapped ϕ_1 with the application of calibration parameters to recover world coordinates. (j) The 3D results colored based on depth value. (k) 3D results with texture mapping applied. (l) Zoomed-in view of the texture mapped results.

4.2. Real-Time 3D Data Acquisition

As previously discussed, three phase-shifted images can be used to reconstruct one 3D frame. Given that DLP projectors project rapidly through the R, G, and B channels to reproduce a color frame, it is natural to encode one phase-shifted pattern into each one of the channels. A computer-generated color-encoded fringe frame, comprised of the three patterns, is sent to the single-chip DLP projector with its color wheel removed. The projector then projects each individual color channel of the image onto the object in sequence and in gray scale. Upon the projection of each channel, the camera is triggered such that each fringe pattern can be captured. After capturing each pattern for one frame, a three-step phase-shifting algorithm can be applied to recover 3D geometry. Furthermore, if the three captured fringe images are averaged together, the result is a single

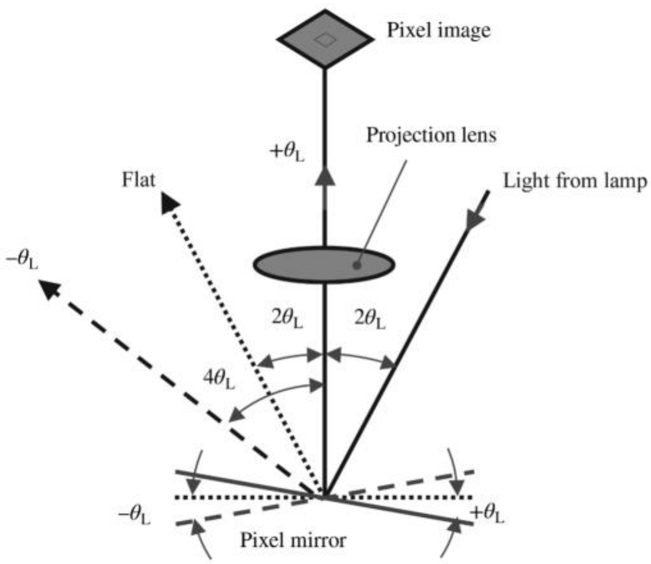


Figure 11. Optical switching principle of a digital micromirror device. (Reprinted with permission from Reference 55. Copyright 2010, Optical Society of America.)

texture image that can be mapped onto the recovered 3D geometry. Based on the application, this texture mapping provides for a more realistic visualization and comes almost computationally free as the data to compute it are already available.

Using a projector with a 120 Hz frame projection rate (360 Hz individual color channel refresh rate), for example, would yield 3D shape measurement speeds of 120 Hz if the camera achieves a fringe capture rate of 360 Hz. Due to this requirement, two projection cycles may be required to capture the three fringe images resulting in a 3D frame acquisition rate of 60 Hz. This is still quite good as real time typically refers to rates of 24 Hz or higher.

4.3. Real-Time 3D Data Processing and Visualization

Although the hardware and fringe acquisition can operate at 60 Hz, for example, computing 3D coordinates for every point for every frame is a computationally expensive task. When left to a single CPU, coordinate recovery at real-time speeds becomes a challenge as the CPU has to compute the coordinate for each point in the phase data sequentially. Given that each point is unique to itself, GPU architecture can be exploited. A GPU, or a graphics processing unit, is a dedicated graphics rendering device specializing in parallel processing. Given this, the GPU can quickly compute each 3D coordinate for each point nearly simultaneously, in lieu of sequentially as with the CPU. Even in computing environments with multiple CPUs, GPUs perform better due to their highly parallelized structure.

After capturing the fringes, phase data can be recovered. This phase map can then be copied from the CPU to the GPU. From here, the GPU can compute the coordinate for each point within the phase data. It can then be used to render the coordinates immediately. The speed advantages that coordinate recovery and visualization via the use of a GPU bring are so great that real-time 3D coordinate calculation can even take place on commercially available PCs that ship with graphics cards (56).

Figure 12 displays an experimental result measuring a live human face using the described structured light technology. On the right is the physical person being captured; on the left is the 3D reconstruction that is happening in real time. In this example, simultaneous 3D data acquisition, reconstruction, and visualization are achieved at 30 Hz with more than 300,000 points per frame being computed and displayed.

5. APPLICATIONS

In today's modern society, imaging technologies are used ubiquitously from the camera on a cell phone being used to

Figure 12. Simultaneous 3D data acquisition, reconstruction, and display at 30 Hz. (Reprinted with permission from Reference 3. Copyright 2010, Elsevier Limited.)

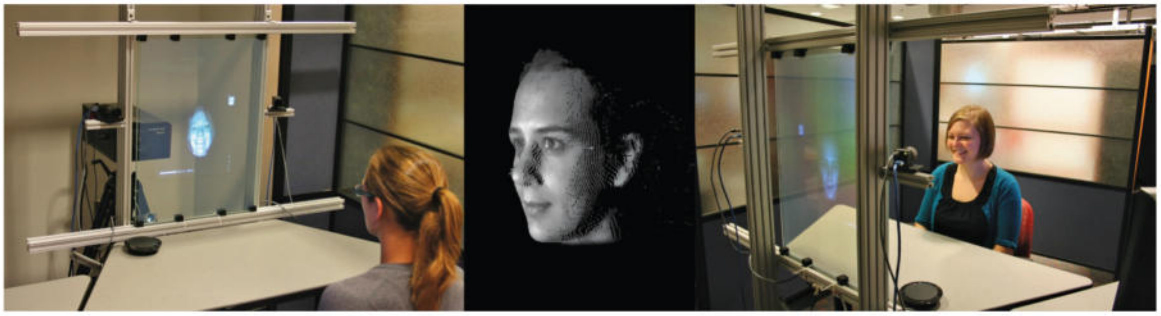


Figure 13. The Portal-s telepresence system enables users to communicate with one another in high-quality 3D over a standard Internet connection in real time.

video chat with a friend across the world; to doctors using an endoscope to see inside of a person to precisely collect cell tissue; to motion-activated security cameras; and to the manufacturing floor where a newly constructed part's quality is assured via measurements. These are a few simple examples, yet value can be perceived in all of them. Now, with high-quality, real-time 3D scanning being possible, objects in the real world can quickly be captured and accurately reconstructed; the third dimension only increases the already wide range of applications for such imaging technologies. This section will highlight examples of applications within the fields of communication, human–computer interaction, medicine and biology, entertainment, security, manufacturing, and remote environment reconstruction where 3D scanning via structured light scanning has been implemented.

5.1. Communication and Collaboration

Societies are now communicating and collaborating with one another on a daily basis from all around the world; however, the technology that makes this possible is often limiting in some manner. For example, when speaking on the phone with someone, the ability to receive and decipher nonverbal communication cues is diminished. Continuing this example, video chat over the Internet now provides users these nonverbal cues, as 2D video is able to provide a higher level of information, which was before missing. As discussed, structured light scanning methods have the capabilities to achieve high-quality 3D scans at real-time speeds; this provides an even greater level of information about the user(s) or environment(s) being captured. At this high fidelity, methods of virtual communication and collaboration may become more realistic.

One such example is the *The Office of the Future* from 1998 (57). To make the office as interactive as possible, the proposed office's three main concepts were the following: the office's equipment (e.g., desks and chairs) could be used as surfaces on which displays were projected, making the office itself a *spatially immersive display*; the geometry of the office's equipment could be captured in 3D and transmitted to a remote location, such as a virtual reality environment, where it could be reconstructed so as to replicate the office; and changes in the office environment could be interpreted so as to provide object tracking, interactive displays, and augmented reality applications. These features aim to increase telecollaboration and effective communication by building "life-like shared-room experience" within existing offices (57). To achieve the concepts, the authors propose that the office's lighting be replaced with cameras and projectors (i.e., structured light systems) to light the room, to project the graphics needed for the interactive displays, and to provide a real-time, comprehensive 3D scan of the room and its objects. The digital representations of the room, its objects, and its inhabitants could then be used to construct virtual meeting rooms with interactive surfaces in an attempt to increase immersion and to increase the level of interpersonal communication.

Another widely accepted technique to improve interpersonal communication is to maintain an appropriate amount of eye contact. Eye contact is nonexistent on telephones and is nearly impossible to achieve in 2D video conferencing systems (as the camera and video displays of the other participant are often not aligned). As previously eluded to, these problems can often be addressed with a higher level of information, in this case the 3D geometry of the participant. A system for achieving eye contact in a 3D video teleconferencing system was developed by Jones et al (58). Their system features the ability to stream 3D video of one remote participant to an audience and stream 2D video of the audience back to the remote participant. To obtain the real-time 3D video feed, a structured light scanner based on Reference 59 is used. The audience views the downsampled feed of the remote participant's face in 3D on an autostereoscopic 3D display; this display is made up of many mirrors that rotate rapidly so as to reflect light so that it appears holographic and 3D in nature to the viewer. The 2D video feed that is used to give the remote participant a view of their audience is also used to locate and track the eye positions of the audience members. These positions are then used to render the 3D geometry of the remote participant's scan so that it appears natural, with eye contact, for each person.

Finally, *Portal-s* (60) is an example of an application in which 3D structured light systems were used to improve communication, this time across the Internet by allowing users to participate in a real-time 3D video teleconference with one another. The goal of a user's Portal-s system was to capture a 3D scan of their face or environment using a structured light scanner, compress the data, and then transmit them to another user over the Internet. The other user's system would then receive the data, decompress them to reconstruct the original 3D geometry, and then

visualize them. Optimally, the second user has a Portal-s system as well, thus allowing each user to view a realistic 3D scan of his/her partner remotely and in real time, as on display in Figure 13.

5.2. Human–Computer Interaction

HCI studies the way users interact with their machines. One obvious way that humans interact and manipulate their machines is with input devices such as a mouse and keyboard. While these input devices work well for manipulating and controlling traditional 2D desktop graphical user interface elements (e.g., windows, toolbars, text boxes), they may not be the best suited to tackle advanced forms of interaction such as manipulating models in 3D space or within virtual reality. Furthermore, those with disabilities may have difficulties operating computers with the traditional devices. For this reason, other forms of input have been studied in search of more natural user interfaces. Using one's face, hands, and body may make certain interactions with computers more natural and may provide methods for disabled users to use systems they might otherwise not be able to. This section explores some of the applications of 3D structured light scanning as it relates to using one's face, hands, and body to interact with machines.

One method for interacting with computing devices that has been studied is hand motion and gestures. Hand gestures especially work well, for example, with manipulating 3D models within virtual reality as the interactions mimic the ones found in the physical world. To do this, there are multiple methods to supply hand input to the computer either via the user wearing a physical glove that tracks the hand or via image processing techniques. The glove approach supplies accurate hand information as input very rapidly, yet the glove itself may be cumbersome and requires the user to wear a physical device. Image processing techniques are able to track the hand, yet may not work well in determining the hand's depth and physical orientation in 3D space. To accurately track a hand in 3D space, it is natural then that 3D information of the hand would be required. In 2003, Guan et al. (61) developed a structured light method to track a user's hand in 3D space so that this information could be used to manipulate objects in virtual and augmented reality. Since then, methods to capture 3D geometry in real time (as discussed in this article) have been developed, thus making hand, and other body part, capture, detection, and tracking possible. Figure 14 shows example frames of hands as captured by a real-time structured light system. To take this concept one step further, Casey et al. (62) used structured light technology to capture and track a user's finger tips in 3D space. This provides a greater level of interaction to potential users as the most minute finger or hand gesture could be detected. Providing accurate input about a user's hand and fingers to a computer in real time opens many new avenues for natural interaction versus using traditional "hands-on" physical input devices such as the mouse and keyboard. It may also provide novel methods for interaction for users who may not easily be able to use traditional devices due to disabilities.

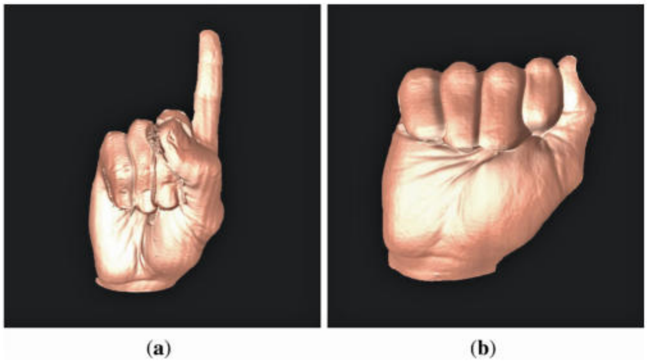


Figure 14. Different examples of minute hand positions and gestures that can be captured by a real-time 3D structured light system; this discrete detail capturing could be applied to any body part. (a) A pointing finger. (b) A curled fist as a result of bringing the pointer finger inward. (Reprinted with permission from Reference (2). Copyright 2014, Wiley.)

Similar to capturing and using one's hand or fingers as input devices, one could also use his/her own face as a means to provide input to a computer. For example, the input could be provided through explicit facial expressions: if a user seems frustrated, then the application may ask the user whether he/she is struck or needs help. Again, image processing or video-based interaction techniques have been devised to track the heads and expressions of a user in 2D space. Such methods may work well in constrained environments and with obvious expressions such as anger, happiness, and sadness, yet they are often limited by a lack of depth information and an inability to decipher the finer features, and therefore emotions, within an image. Obtaining highly accurate 3D geometry in real time, however, presents a solution to some of these drawbacks.

Sandbach et al. (63), for example, detailed how dynamic 3D facial expressions can be recognized. With a higher fidelity of information available, a wider range of expressions can be classified and recognized (e.g., puckered lips or a clenched jaw). Figure 15 shows an example of 3D frames captured from a real-time structured light scanner. Figure 15a shows a simple smirk and Figure 15b a raised brow with a wrinkled forehead. These expressions could then be used by users to interact with their machines either explicitly or implicitly. In the explicit case, a user could map some expression to some action within an application or within their computing environment. In the implicit case, facial expressions can be correlated to user emotions. For example, if an application used a user's 3D geometry to detect a frown or a clenched jaw, it could deduce that the user may be frustrated in his/her use of the software. The application could then present the user with help options. These are simple examples, yet it is clear that easily recognizable 3D facial expressions provide users a natural method to interact with their machines over traditional input devices.

The search for natural user interfaces has been ongoing since computing itself began. The driving idea is that if systems are easy to understand and use then more people will use and benefit from them. The capability to capture, track, and analyze high-quality 3D geometry information at real-time speeds then has great implications when it comes

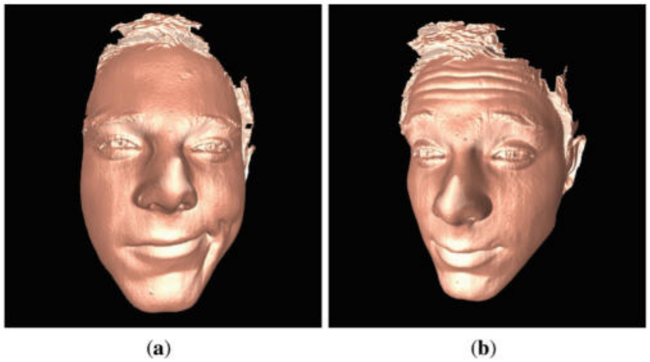


Figure 15. Different examples of minute facial features and gestures that can be captured by a real-time 3D structured light system. (a) A smirk could be captured and used as a control. (b) Although a very discrete feature, wrinkles can easily be seen on the forehead and can be deciphered. (Reprinted with permission from Reference (2). Copyright 2014, Wiley.)

to developing such natural interfaces, especially when it comes to developing novel methods of interaction for those with disabilities.

5.3. Entertainment

The previous section discussed examples of applications using structured light technology to create new, more natural methods for users to interact with their machines. These new methods are not constrained only to increasing ease of use and efficiency for a user. Structured light technology has been used within the entertainment industry, as well.

The *Microsoft Kinect*, for example, is a 3D sensing device that uses structured light techniques to construct a depth map in real time within its hardware. The Kinect then provides the depth map to an application to make use of. The application can also use the Kinect's algorithms to decipher features within the data, such as gesture recognition, skeletal detection and tracking, pose detection, and user segmentation (64). Many video games were developed for and the Xbox 360 which allowed or more users to use their bodies as input devices into the games they were playing. For example, in *Kinect Sports* users can use their bodies, arms, legs, and hands to play a variety of sporting simulations, including soccer, volleyball, bowling, and table tennis. Playing games with one's own body via a 3D structured light sensor, versus a hardware controller device, is an attempt to make the entertainment experience feel more natural.

Not only can interactions within video games be more natural, but they can also better represent physical reality via the use of 3D structured light technologies. Zollhöfer et al. (65) used a Kinect sensor to capture geometry and color texture of a user's face. The face scan is then fitted to a facial model and can be applied to an in-game avatar, thus allowing the users to be accurately represented as themselves within the game they are playing.

Structured light scanners have also been used to provide high-quality 3D data for use within multimedia productions. For example, Zhang used a system to capture data for the rock band *Radiohead* to produce their "House of Cards" music video (66); a screenshot of the music video is shown in Figure 16. Zhang's structured light technology was also used by Zaftig Films to produce the 2015 film *Focus* starring Will Smith.

5.4. Medicine and Biology

2D imaging technologies have extensive applications in the area of biology and medicine. For instance, surgical imaging devices, such as endoscopes, are widely adopted

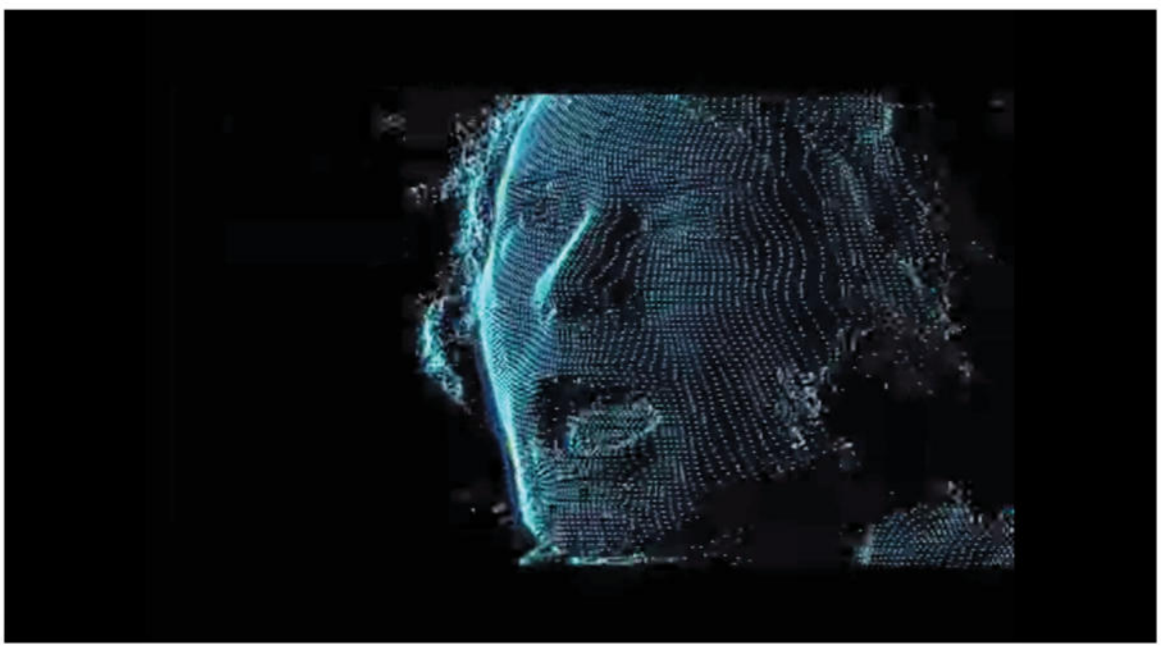


Figure 16. A screenshot from Radiohead's "House of Cards" music video (66). A 3D structured light scanner was used to capture the raw data.

in surgery rooms that help doctors locate and navigate affected organs. Moreover, imaging technologies are also widely implemented in cosmetic surgery and in the diagnosis of eye-related diseases. However, 2D imaging can sometimes be deceiving since it does not provide doctors with any depth information. In recent years, scientists have been seeking to implement 3D imaging in the field of biomedical science. Among all 3D measurement technologies, the structured light system is one of the promising approaches to achieve high-quality real-time 3D imaging due to its flexibility, speed, and accuracy.

Facial Anthropometry. Imaging 3D facial expression could be beneficial for cosmetic surgery or identifying some diseases such as facial paralysis. By analyzing 3D data in different frames of facial expressions, doctors could essentially carry out diagnosis by matching the facial motion with some known patterns of illness.

For example, Bhatia et al. (67) initiated a structured light scanner for quantifying facial surface change. By using six pairs of camera–projector systems, their system has realized 360° coverage of a subject's full head. This research was specifically carried out for detecting facial geometry changes after subcutaneous fluid injection. Mehta et al. (68) presented a method to quantitatively diagnose facial paralysis through tracking some critical markers on the face (e.g., mouth corners) using 3D facial video data of patients. They used a visible 3D video system developed by Zhang and Yau (42) to capture facial expressions at 60 Hz. Olesen et al. (69) have incorporated a dual-camera structured light system into a PET scanner, which has been tested in a clinical setting for 3D head motion tracking. A near-infrared (NIR) video projector was used to create a more comfortable setting for patients. They have proved that the accuracy of this system is comparable to a commercial optical tracking system with root mean square (RMS) errors of 0.09° for ±20° axial rotation and 0.24 mm for ±25 mm translation.

Cardiac Mechanics Analysis. Scientists have devoted great efforts to study the functional mechanism of the heart and heart-related diseases. Structured light methods have been used to capture heart motion for cardiac mechanics study. This is because 3D images of the cardiac surface could assist the biomedical scientists to better visualize and quantify the geometric motion and thus to understand the functional conditions of the heart.

Iyengar et al. (71) adopted structured light imaging technology for bovine heart valve leaflet motion analysis. A diode laser was used to illuminate the structured light pattern, while a stereo borescope was used for image acquisition. It realized dynamic imaging of heart valve leaflet motion while maintaining a high temporal and spatial resolution. This research can be applied to the studies of basic heart valve mechanics and bioprosthetic heart valve-related issues. Laughner et al. (70) examined the surface mechanics of a live rabbit heart with high-speed structured light imaging. The structured light system used in this research has an in-plane spatial resolution of 87 μm and a depth resolution of 10 μm with a sampling speed of 2000 Hz. By adopting a nonrigid motion tracking algorithm based on isometry-maximizing optimization, the correspondence between consecutive 3D frames can be determined and thus the surface mechanics can be known. They have achieved a mean tracking error of 0.37 ± 0.10 mm and a maximum error of 1.23 mm for all measured samples. Figure 17 shows an example of surface mechanics of a rabbit heart.

3D Endoscopy Imaging. The endoscope is now a widely used surgical imaging device. It is often used in minimally invasive surgery. Given recent advances within pathology, traditional 2D endoscopic imaging may no longer fulfill the needs of surgical planning and practices. Therefore, scientists have been discovering new methods to realize 3D endoscopic imaging capabilities. One of the popular approaches for producing a 3D endoscope is by means of stereo vision. Essentially, the object surface is imaged by two different viewpoints with a known baseline separation, and the 3D geometry is reconstructed through triangulation. Over the years, a number of stereo vision-based endoscopy research works have been carried out (72–79). Since stereo algorithms are based on feature detection, it is difficult for them to achieve high accuracy if the surface does not have strong texture variations (e.g., heart surface).

Scientists are also exploring the feasibilities of developing a 3D endoscope using structured light methods. To transfer the structured light 3D imaging technology into an endoscope, it is crucial to reduce the size of the imaging devices (camera and projector) so that a miniaturized probe can be produced. Therefore, it is natural to think of inserting the illumination device into the working channel of endoscopes. Schubert and Müller (80) first initiated a structured light 3D bronchoscopy that was used to measure hollow biological organs. A laser fiber probe that illuminates a ring of laser light was inserted into the instrument channel of the endoscope, and the inner surface geometry was reconstructed by analyzing the deformation of the emitted ring. Since only one ring of laser light is projected at a time, the endoscope has to move to cover the entire surface geometry. As a variation of this approach, Armbruster and Scheffler (81) proposed a new type of endoscope that measures 3D shape by analyzing the deformations of several projected lines, and the measurement accuracy was improved by adopting a phase-shifting algorithm. Kemper et al. (82) inserted an electronic speckle pattern interferometer (EPSI) into an endoscope camera system and achieved a data acquisition rate of 25 Hz; such a system can be used for *in vivo* minimal invasive diagnosis. This compact and flexible endoscopic design is applicable to hand-held examinations in minimal invasive surgery. Bendall et al. (83) proposed a concept of miniaturizing the phase-shifting pattern projection system. Through incorporating multiple controllable light emitters and a fringe pattern mask into the endoscope system, the sinusoidal pattern coding strategy shown in Section 2.6 can be realized. This conceptual design has the potential to increase the computational efficiency by enabling surface scanning. Recently, Schmalz et al. (84) proposed a prototype system that incorporates a catadioptric camera and a slide projector into a single probe. The projector projects a pseudo

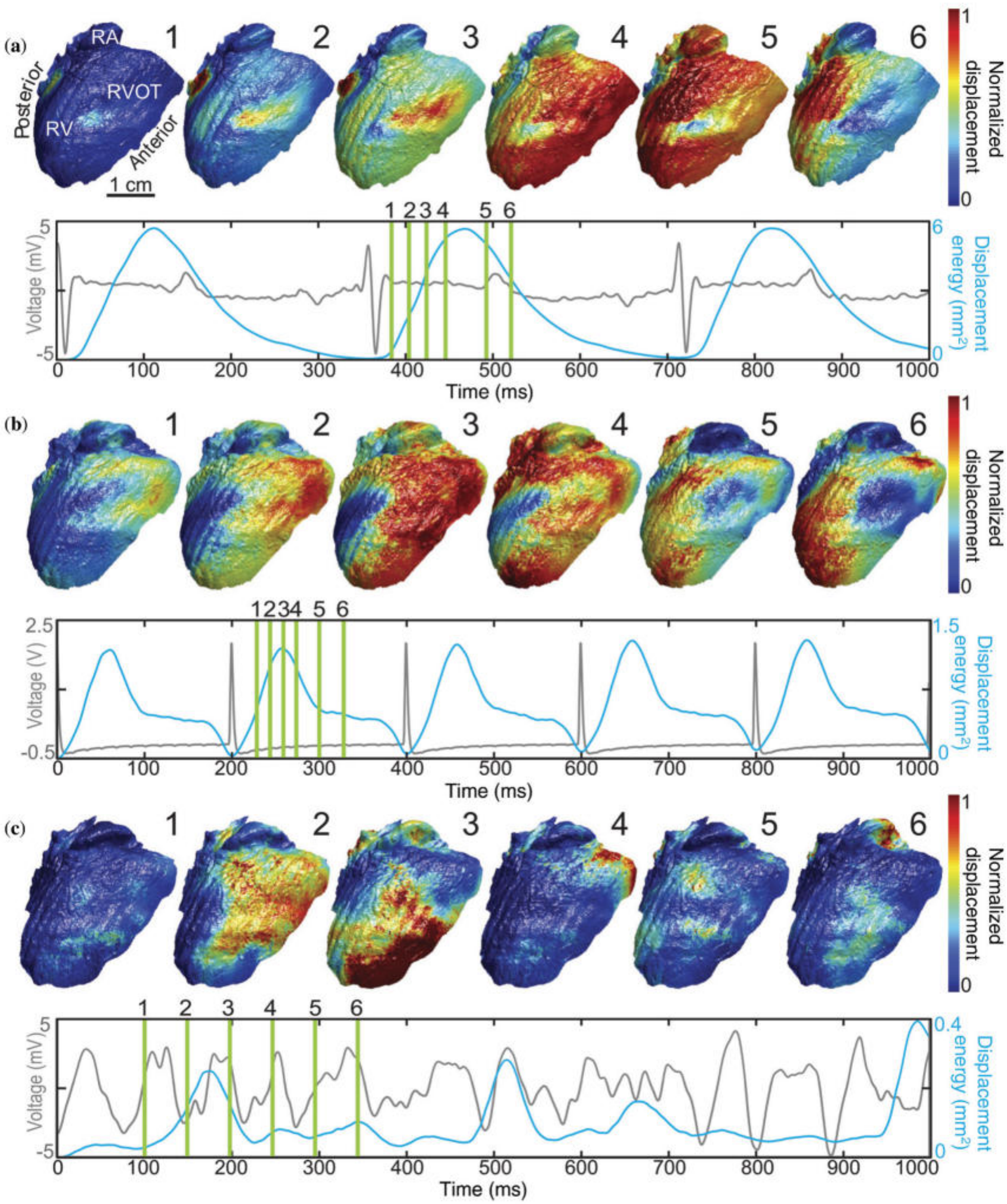


Figure 17. Analysis of motion in the beating rabbit heart for SR (a), apical pacing (CL = 200 ms; b), and ventricular fibrillation (c). For each case, sequences of six normalized displacement maps are displayed. Location of sequence frames is indicated by green bars on a simultaneously recorded ECG (gray). Displacement energy (blue), a measure of total tissue displacement, is overlaid on each respective ECG. (Reprinted with permission from Reference (70), Copyright 2012, American Physiological Society.)

random array of color rings that can be captured and analyzed to generate 3D shape through triangulation. This system has realized a 3D data acquisition rate of 30 Hz and a measurement accuracy of 0.1 mm.

It is also possible to realize image acquisition and structured light illumination in different probes, which improves the flexibility of the system design, albeit the calibration of such system becomes more challenging. Fuchs et al. (85) designed a separate projection probe from a regular laparoscope using structured light illumination. A video projector and some associated optics were used to generate structured light pattern stripes. The image acquisition was performed using a regular laparoscope. The probe was placed in parallel with the laparoscope with a fixed baseline separation. This approach has demonstrated its promise for real-time 3D imaging, despite its limitations of slow depth extraction and limited calibration accuracy. Maurice et al. (86) have designed a 3D laparoscope

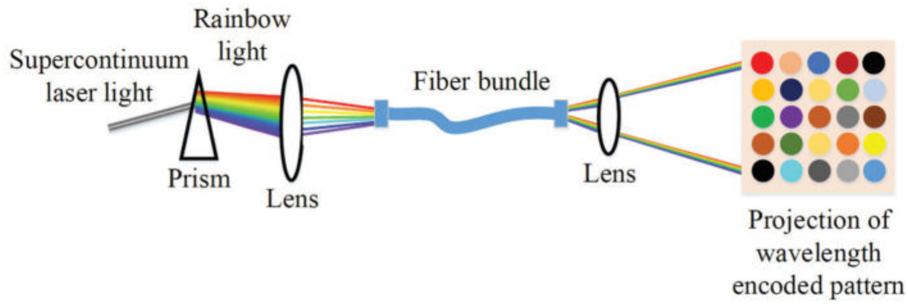


Figure 18. Schematic diagram of 3D endoscope using spectral encoded structured light.

by connecting a camera and a miniaturized projector to two separate endoscopes. A monochromatic coded pattern was designed with locally unique features, and the technique called spatial neighborhood scheme was used for feature correlation. This 3D laparoscope has already been used in the operation room for minimally invasive situations. It has reported to produce reasonable measurement results at an image acquisition rate of 25 Hz. Reiter et al. (87) designed another structured light system with side-by-side laparoscopes. Different from the previous approach, a De Bruijn sequence pattern, as introduced in Section 2.1, was illuminated in the projection channel. It has reported an accuracy of 0.20 mm with a measuring distance of 17.50 mm.

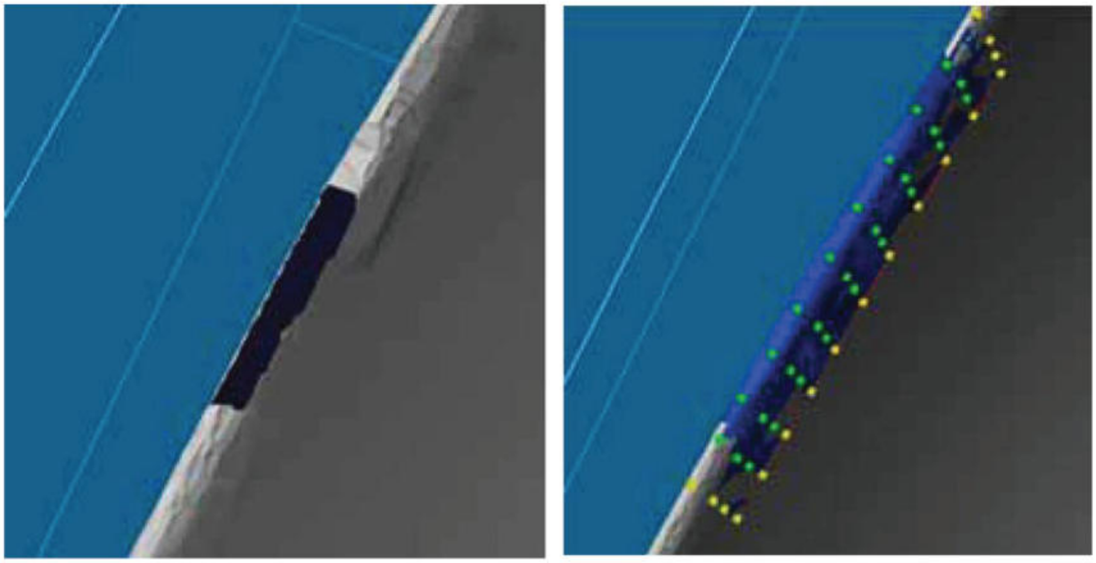
Apart from using a miniaturized projector as a projection device, another popular approach is to use a fiber array to illuminate the structured light pattern because of the small size nature of fibers (e.g., diameter in μm). Clancy et al. (88) designed a special probe with fiber bundles that encodes each projected point with a specific wavelength. The principle of this method is illustrated in Figure 18. The spectrally encoded structured light pattern is captured by a video camera, and then the 3D surface is reconstructed based on the relationship between the light wavelength and the RGB color space. The coding strategy of this technology is comparable to the principles introduced in Section 2.1, in which each point within the whole scene is uniquely coded. To enhance the computational efficiency, a different type of structured light endoscope can be developed by adopting continuous sinusoidal phase codifications (see Section 2.6). The enabling technology is the use of fibers and laser to generate sinusoidal patterns (89). The 3D reconstruction algorithm is similar to a digital fringe projection system while reducing the size of the projection system to μm level. This technology enlightens a future direction for 3D endoscope development.

5.5. Biometric Authentication

Biometric measurements are being used for security and authentication purposes worldwide within airports, shopping centers, banks, and homes due to their perceived uniqueness. Whereas a short PIN code or passphrase can be cracked or forgotten, biometric measurements are often unique to the user and are difficult to alter. Some common biometric measurements that are performed for different security and authentication purposes are iris, fingerprint, retina, voice, and facial recognition (90). For example, many modern laptops now feature account login capabilities via detecting and recognizing a user's face within a webcam's 2D video feed who has predefined access to the device. Although such features offer convenience and security via a perceived unique form of authentication, they are still often faulty by not being able to detect the user or by falsely identifying another person as the user. If the authentication does work, however, authentication may be gained by a malicious user simply by holding a photograph of the privileged user in front of the webcam, for example. In general, the inaccuracies and security holes surrounding features implemented from 2D video analysis are due to a lack of information in the data. Some experts have proposed conjectures that state that the 3D shape of one's face is more powerful for facial recognition than one 2D image (91). With the introduction of a third dimension then, these security systems may become more accurate and therefore may not be as easily deceived.

In the aforementioned example, it was noted that traditional facial recognition systems can easily be deceived by placing a photo of the user in front of the camera. Lagorio et al. (92), however, developed a biometric authentication method to determine whether the user in front of the capturing system was indeed physically present and not just a photograph or video. They developed a simple *liveness detection* method that can decipher whether the object in front of a 3D scanner is geometrically representing a face or simply a flat object (such as a photo or video of the user). The 3D liveness detection method could then be used alongside traditional 2D facial recognition to better ensure that the privileged user is indeed the one trying to access the system at that moment. There are numerous other approaches to increasing the facial recognition capabilities of 2D systems with the inclusion of 3D data; these are typically referred to multimodal 3D + 2D face recognition approaches (93).

Facial biometric authentication methods can be improved by using high-quality 3D data, such as data captured with a structured light scanner, to perform the facial recognition and matching. For example, Drira et al. (94) developed a framework to recognize 3D faces under different expressions, occlusions, and poses. They first capture a 3D scan, fill any holes within the geometry, crop the face out from the rest of the scan (boundaries, hair, torso, etc.), and then smooth the geometry. Radial facial curves are then measured outward from the tip of the nose. These radial curves are then used to define a shape metric for the



(a) (b)

Figure 19. Example process for defect repair: (a) acquired point cloud with defect from 3D scanning; (b) defect-free model obtained from NURBS fitting. (Reprinted with permission from Reference (101), Copyright 2006, Elsevier Limited.)

facial surfaces. The derived shape metric for one 3D scan of a person's face is then matched within a database of 3D scans; if a match is found, then recognition has occurred. The framework detailed by Drira et al. was able to achieve competitive results relative to the state of the art for 3D face recognition even when testing the framework on data sets containing occlusions, nonfrontal views, and large facial expressions (94).

Iris recognition is another form of contact less biometric authentication in which users scan their eyes to gain access to a virtual or physical system (95). The iris recognition techniques work quite well; however, their perceived security has been decreasing in recent years due to the development of inexpensive and highly customizable contact lenses. These lenses could be plausibly used to falsely gain access into a system via an iris scanner. For this reason, Connell et al. (96) developed a structured light scanning system to obtain 3D geometry of a user's eye. The scan then was used to detect the presence of an artificial item, such as a custom-printed contact lens, which was on top of the real iris. Their experimental results were successful in detecting whether the scan of the eye was a naked eye or whether the naked eye was obscured by a patterned contact. Their method was also able to decipher well between a user wearing a standard contact and a contact with a pattern printed onto it. Traditional 2D iris recognition scanners could easily be outfitted with this technology, as well; all that would need to be added is a projection unit. Whereas custom-printed contact lens may allow an unprivileged user access to a system protected with a 2D iris scanner, an improved system that utilizes the high fidelity of the 3D information made available by structured light technologies may provide a greater level of security. Similar to the multi modal 3D + 2D face recognition approaches, these techniques would then be multi modal 3D + 2D iris recognition approaches. Some modern approaches to biometric authentication have also been looking into authentication using multiple forms of biometric readings. For example, one multi modal biometric authentication technique that was developed by De Marsico et al. (97) used both facial and iris recognition to recognize users of mobile devices.

5.6. Manufacturing

3D optical scanning can also be applied to the manufacturing area for dimensional measurements that can be used for reproduction of the parts, defect detection, repairs, and further quality assurance.

Reverse engineering (RE) is another area that can benefit from using structured light technologies. RE typically refers to the process of reproducing the manufacturing details of an existing object (98). In the past decade, many attempts have been made to incorporate structured light 3D scanning into a reverse engineering system. Carbone et al. (99) initiated a type of reverse engineering system by combining a structured light vision system with a coordinate measuring machine (CMM). The object surface was first scanned by a structured light system to obtain a "rough" knowledge of the object geometry. This prior knowledge of the object geometry was utilized to optimize the programming efficiency for CMM by reducing the number of touching points and iterations. The final modeling was completed through this refined measuring process with the CMM. The measurement error in depth of this technology was reported to be within 0.1 mm for a height range of 100 mm. Li et al. (100) proposed to design a complete reverse engineering system for the rapid manufacturing of complex objects. Different from Carbone's method, the 3D modeling was completely determined by the structured light system, and the object is reconstructed with multi view scanning. The whole modeling process is composed of 3D acquisition,

multi view data registration, surface integration, and iso surface extraction. The models are exported in STL format that finally goes to FDM 2000 machine to manufacture products.

RE plays an important role nowadays in product life cycle management by combining RE with the additive manufacturing methods. As a flexible and accurate tool, structured light 3D imaging has good potential to be used in the whole design process. It usually starts with measuring the 3D geometry of an existing part if the digital design is not available; the measured 3D data can be refined and modified on a computer to reflect the changes needed from the existing parts; the modified 3D representation is then sent to the manufacturing equipment to fabricate the part; and the fabricated parts can then be scanned again by the 3D scanning system for another cycle.

3D scanning can also be used to recover a damaged or worn part. Park and Chang (102) proposed a structured light-based system to fix missing parts. This research aims at resolving missing areas by automatically computing the next scanning orientation until the whole 3D surface is obtained. It has demonstrated a good computational efficiency for determining the next scanning orientation (usually computed within several seconds), which proves that the whole process can be completed within a reasonable amount of time. Gao et al. (101) presented a reverse engineering system for repairing a worn part. The whole process starts by measuring the 3D geometry of the worn part, as shown in Figure 19a. To create a defect-free model, it replaces the triangular meshes of the detected defects areas (bumps, dents, and holes) with NURBS or Bezier fitted surface, as shown in Figure 19b. Then the repaired patches were extracted for welding and laser cladding.

Due to its speed and accuracy, structured light technologies can also be used in the manufacturing industry for inspection. Industrial inspection generally refers to the activities of measuring or testing certain characteristics of a manufactured object, which could include dimension measurements or defect detection. In 1985, Jalkio et al. (103) designed a 3D inspection system using multistripe structured light. This method essentially projects several laser lines and reconstructs the 3D shape using triangulation. Despite its limited accuracy, this early stage of structured light inspection system has indicated a novel direction of performing non contact 3D inspection. Zhang and Wei (104) improved the accuracy of this approach by adopting a backward propagation neural network method for calibration; it has reached a testing accuracy of 0.128 mm. Gühring (105) started to apply the structured light method to part measurement using a commercial projector, which has further improved the flexibility of the system setup. In recent decades, with the advancements in system calibration accuracy, researchers have implemented structured light 3D scanning to various kinds of applications. Bieri and Jacot (106) designed a prototype 3D prototype system for quality control in a production line. By creating a random phase noise estimator from different noise sources, they have reduced the depth measurement uncertainties of a typical structured light 3D scanner. It has achieved a systematic error of 43 μm. Li et al. (107) presented another application of a structured light method in detecting the defects of the weld bead with online vision inspection. After 3D reconstruction of the geometry, some feature extraction algorithms were used to extract the properties of the welding surface, such as groove width, weld bead width, and filling depth. The defects such as plate displacement, weld bead misalignment, and undercut can be detected online. Xu et al. (108) designed a real-time 3D shape inspection system for automotive parts using structured light pattern projection. The system calibration was performed under a known height flat gauge that has achieved a system accuracy of 0.182 mm. It has reported some reasonable measurement results of a car's pillar, door, and windshield. Usamentiaga et al. (109) designed a special structured light sensor using two laser stripes for 3D inspection with the presence of vibration. To perform vibration-resistant 3D imaging, a second laser stripe was added with a time offset to compensate for vibration effect during the scanning process. The accuracy of the 3D imaging system was claimed to reach up to 66 μm with the presence of vibration noise.

Furthermore, with the recent rapid improvements of 3D scanning and data processing speeds, it might be possible to embed a 3D imaging system on the production line for *in situ* quality control, or to provide feedback to the machine for closed-loop controls.

5.7. Remote Environment Reconstruction

The hardware that makes up a given structured light system is intelligently selected to best suit the application in question. Given this, structured light systems are very versatile. One more recent trend in which structured light scanners are being used is in remote environment reconstruction. Given that 3D structured light methods are contact less in nature, they are well suited to remotely capture geometry in environments where it may be too dangerous or simply impossible for humans to physically measure themselves.

One such scenario is underwater 3D geometry reconstruction. Until recently, most research into underwater 3D reconstruction focused on long-range applications such as mapping the sea floor. In 2011, however, Bruno et al. demonstrated experimental results of underwater 3D reconstruction at a close range using a structured light scanner (110). The experiments conducted captured and measured different objects submerged in water with different turbidity levels. The system developed by Bruno et al. was built with commodity hardware and was not very large, as well. Given this, an extension of this work would be to outfit deep sea divers, submarines, or other underwater vehicles with portable structured light systems. The systems could then be used to remotely study, at a relatively high accuracy in 3D, the growth of coral reefs or reconstruct archaeological artifacts that may be difficult to recover, for instance.

Another example of remote environment takes place within railway tunnels where inspection is needed to ensure safe operation. In the past, both contact and contact less methods had been tried to measure and inspect railway tunnels in 3D. The contact methods tried were simply too slow and labor intensive. Several contact less

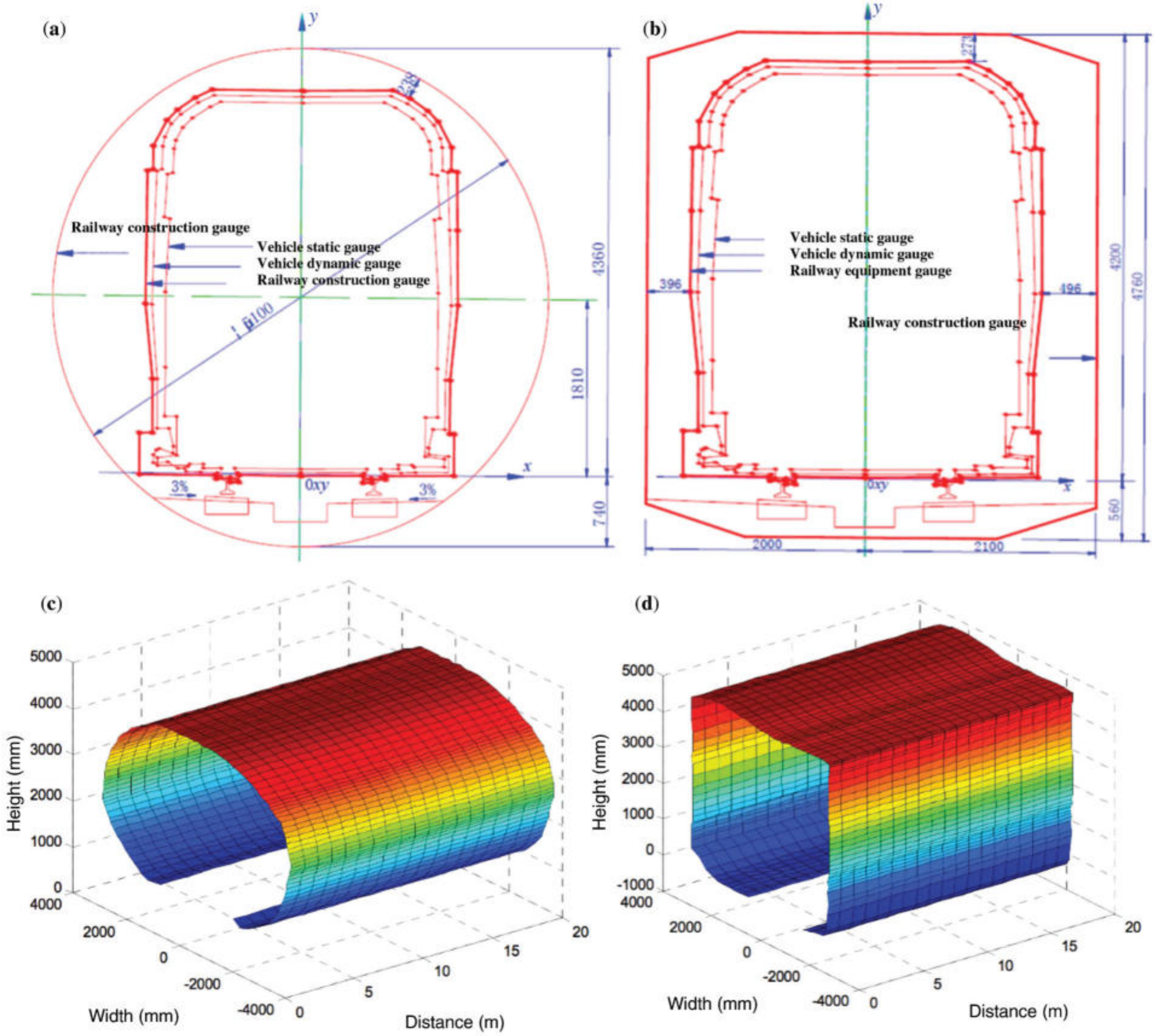
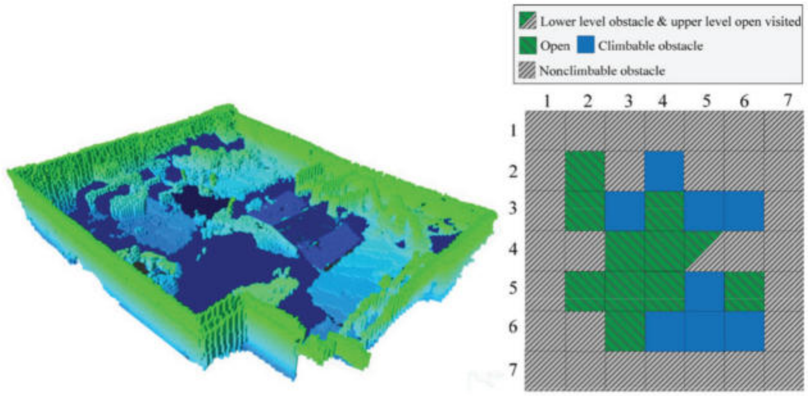


Figure 20. (a) Circular tunnel blueprint. (b) Rectangular tunnel blueprint. (c) Circular tunnel dynamic measurement results. (d) Rectangular tunnel dynamic measurement results. (Reprinted with permission from Reference (111), Copyright 2015, Multidisciplinary Digital Publishing Institute.)

methods were tried, such as those using time-of-flight inspection and stereoscopic vision, yet these technologies could not meet the high-speed and high-accuracy requirements needed for the on-site inspections. Given this scenario, Zhan et al. (111) developed a multi camera and structured light vision system to provide a complete field of view of railway tunnels in 3D at high speeds with a high accuracy. This system was then fitted to a vehicle and collects each cross-sectional profile of the railway tunnel with an interval of 250 mm at a speed near 60 km/h (111). Their equipment and algorithms were able to achieve full field-of-view clearance inspection of a railway tunnel in 3D with a maximum measurement error of -1.47 mm, which is well within the required accuracy of 5 mm. Figure 20 shows recovered 3D data of a railway tunnel versus the tunnel's original drawing. Given this new technology based on structured light, railway inspections may now be more accurate and can be performed at a greater frequency, as they are fast and easy to perform, in an attempt to prevent adverse incidents.

In not all situations, however, are disasters preventable through advanced inspection techniques; take natural disasters, for example. Although natural disasters are impossible to prevent, the course of action taken after an incident can save lives and property. For this reason, 3D structured light scanners are being used to gather high-quality, real-time 3D assessments of property damage so as to support decision making for search and rescue as well as disaster relief. A 2014 study performed by Qiu et al. (112) explored the feasibility of using structured light technologies to perform 3D damage assessments after natural disasters. The study performed measurements of a car and a house so as to simulate what types of objects might be measured after a natural disaster scenario. The results presented were accurate enough to visually decipher whether damage had been done to the object or structure. To enable a more safe and rapid analysis of disaster-affected areas, structured light scanners could then be fitted to unmanned aerial vehicles



(a) (b)

Figure 21. (a) The 3D Octomap of a remotely and cooperatively reconstructed urban environment. (b) The Octomap's corresponding occupancy grid. (Reprinted with permission from Reference (113), Copyright 2015, Wiley.)

(UAVs) or remotely operated rescue robots. Assessment of damage in real time at high accuracies can then be performed *in situ* with aims to save lives and impact decision making.

Similarly, Liu and Nejat have developed methods for the semiautonomous control of multiple robots within urban search and rescue operations (113). Outfitted with different devices (e.g., depth cameras, 2D cameras, and thermal cameras), multiple rescue robots can be used to remotely reconstruct their unknown, urban environments in 3D through cooperative exploration. As the reconstructed environment contains 3D geometry, different metrics can be measured and used in the construction of a corresponding *occupancy grid*. Examples of grid occupancies that are extracted from the shared 3D geometry and other sensed data are open areas, climbable obstacles, non climbable obstacles, potential victims, and lower level obstacles (113). This occupancy grid can then be used by the robots to perform path planning during search and rescue operations. Figure 21 shows an example of a reconstructed 3D environment and its corresponding occupancy grid.

6. SUMMARY

This article has covered 3D scanning via structured light techniques. The general principles behind several different approaches and methods to obtain real-time 3D scanning have been introduced. Fields in which high-quality 3D structured light scanners are being utilized were presented along with several specific applications. To further increase the rate of adoption for structured light technologies, the size and robustness of the systems can be improved. Effort has started and will continue to miniaturize the system's components while maintaining a high quality of data, as well as increase the robustness of the system. Furthermore, the cost of the hardware must continue to decrease for this technology to become universally available. If these remaining challenges can be addressed, such that this technology may become generally accessible, the potential applications in which high-quality 3D data can be used will expand immensely with the possibility to further impact the fields of communications, biology, medicine, manufacturing, human–computer interaction, robotics, and security.

It is important to note that this article, by no means, covers all structured light methods that have been developed and utilized, or all areas that structured light methods could be applied to. This article serves as a reference for readers to understand some of the popular methods and to learn about some of the application areas. Readers can access other general reviews (1,6,7) and, if further interested, can learn more in-depth details about specific topics from the individual references used throughout this article.

ACKNOWLEDGMENTS

The authors would like to thank fellow lab member Zexuan Zhu for capturing the data shown in Figure 10. This study was sponsored by the Fundamental Research Funds for the Central Universities and the National Science Foundation (NSF) under Grant Nos. CMMI-1150711 and CMMI-1300376. Any opinion, findings, and conclusions or recommendations expressed in this article are those of the authors and do not necessarily reflect the view of the NSF.

BIBLIOGRAPHY

1. S. Zhang. *Handbook of 3D Machine Vision: Optical Metrology and Imaging*. CRC Press, 2013.
2. T. Bell, N. Karpinsky, and S. Zhang. *Real-Time 3D Sensing with Structured Light Techniques*. Wiley, 2014; pp 181–213.
3. S. Zhang. Recent Progresses on Real-Time 3D Shape Measurement Using Digital Fringe Projection Techniques. *Opt. Lasers Eng.* **2010**, *48*, pp 149–158.
4. S. Lei and S. Zhang. Flexible 3-D Shape Measurement Using Projector Defocusing. *Opt. Lett.* **2009**, *34*, pp 3080–3082.
5. S. Zhang, D. Van Der Weide, and J. Oliver. Superfast Phase-Shifting Method for 3-D Shape Measurement. *Opt. Express* **2010**, *18*, pp 9684–9689.
6. J. Salvi, S. Fernandez, T. Pribanic, and X. Llado. A State of the Art in Structured Light Patterns for Surface Profilometry. *Pattern Recogn.* **2010**, *43*, pp 2666–2680.

7. J. Salvi, J. Pages, and J. Batlle. Pattern Codification Strategies in Structured Light Systems. *Pattern Recogn.* **2004**, *37*, pp 827–849.
8. U. R. Dhond and J. K. Aggarwal. Structure from Stereo: A Review. *IEEE Trans. Syst. Man Cybern.* **1989**, *19*, pp 1489–1510.
30. J. Davis, R. Ramamoorthi, and S. Rusinkiewicz. Spacetime Stereo: A Unifying Framework for Depth from Triangulation, in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*; Vol. 2, IEEE, 2003; p II-359.
10. L. Zhang, B. Curless, and S. M. Seitz. Spacetime Stereo: Shape Recovery for Dynamic Scenes, in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*; Vol. 2, IEEE, 2003; p II-367.
11. L. Zhang, N. Snavely, B. Curless, and S. M. Seitz. Spacetime Faces: High-Resolution Capture for Modeling and Animation. In *Data-Driven 3D Facial Animation*; Springer, 2008; pp 248–276.
12. B. Li and S. Zhang. Structured Light System Calibration Method with Optimal Fringe Angle. *Appl. Opt.* **2014**, *53*, pp 7942–7950.
13. B. Carrihill and R. Hummel. Experiments with the Intensity Ratio Depth Sensor. *Comput. Vis. Graph. Image Process.* **1985**, *32*, pp 337–358.
14. G. Chazan and N. Kiryati. Pyramidal Intensity-Ratio Depth Sensor. Center for Communication and Information Technologies, Department of Electrical Engineering, Technion – Israel Institute of Technology, 1995.
15. Q. Fang. Linearly Coded Profilometry with a Coding Light that Has Isosceles Triangle Teeth: Even-Number-Sample Decoding Method. *Appl. Opt.* **1997**, *36*, pp 1615–1620.
16. Q. Fang and S. Zheng. Linearly Coded Profilometry. *Appl. Opt.* **1997**, *36*, pp 2401–2407.
17. P. Jia, J. Kofman, and C. English. Two-Step Triangular-Pattern Phase-Shifting Method for Three-Dimensional Object-Shape Measurement. *Opt. Eng.* **2007**, *46*, pp 083201-1–083201-9.
18. K. Wu, J. Xi, Y. Yu, and Z. Yang. 3D Profile Measurement Based on Estimation of Spatial Shifts Between Intensity Ratios from Multiple-Step Triangular Patterns. *Opt. Lasers Eng.* **2013**, *51*, pp 440–445.
19. P. S. Huang, S. Zhang, and F.-P. Chiang. Trapezoidal Phase-Shifting Method for Three-Dimensional Shape Measurement. *Opt. Eng.* **2005**, *44*, pp 123601-1–123601-8.
20. C. Zhang, P. S. Huang, and F.-P. Chiang. Microscopic Phase-Shifting Profilometry Based on Digital Micromirror Device Technology. *Appl. Opt.* **2002**, *41*, pp 5896–5904.
21. P. S. Huang, C. Zhang, and F.-P. Chiang. High-Speed 3-D Shape Measurement Based on Digital Fringe Projection. *Opt. Eng.* **2003**, *42*, pp 163–168.
22. M. Takeda and K. Mutoh. Fourier Transform Profilometry for the Automatic Measurement of 3-D Object Shapes. *Appl. Opt.* **1983**, *22*, pp 3977–3982.
23. H. Schreiber and J. H. Bruning. Phase Shifting Interferometry. In *Optical Shop Testing*, 3rd ed.; Wiley, 2006; pp 547–666.
24. D. Malacara. *Optical Shop Testing*, Vol. 59. Wiley, 2007.
25. D. C. Ghiglia and M. D. Pritt. *Two-Dimensional Phase Unwrapping: Theory, Algorithms, and Software*, Vol. 4. Wiley, 1998.
26. G. Sansoni, M. Carocci, and R. Rodella. Three-Dimensional Vision Based on a Combination of Gray-Code and Phase-Shift Light Projection: Analysis and Compensation of the Systematic Errors. *Appl. Opt.* **1999**, *38*, pp 6565–6573.
27. S. Zhang. Flexible 3D Shape Measurement Using Projector Defocusing: Extended Measurement Range. *Opt. Lett.* **2010**, *35*, pp 931–933.
28. O. Hall-Holt and S. Rusinkiewicz. Stripe Boundary Codes for Real-Time Structured-Light Range Scanning of Moving Objects, in *8th IEEE International Conference on Computer Vision*; 2001; pp II:359–II:366.
29. Y. Wang, S. Zhang, and J. H. Oliver. 3D Shape Measurement Technique for Multiple Rapidly Moving Objects. *Opt. Express* **2011**, *19*, pp 5149–5155.
30. Q. Z. X. Su, L. Xiang, and X. Sun. 3-D Shape Measurement Based on Complementary Gray-Code Light. *Opt. Lasers Eng.* **2012**, *50*, pp 574–579.
31. Y. Wang and S. Zhang. Novel Phase Coding Method for Absolute Phase Retrieval. *Opt. Lett.* **2012**, *37*, pp 2067–2069.
32. C. Zhou, T. Liu, S. Si, J. Xu, Y. Liu, and Z. Lei. Phase Coding Method for Absolute Phase Retrieval with a Large Number of Codewords. *Opt. Express* **2012**, *20*, pp 24139–24150.
33. C. Zhou, T. Liu, S. Si, J. Xu, Y. Liu, and Z. Lei. An Improved Stair Phase Encoding Method for Absolute Phase Retrieval. *Opt. Lasers Eng.* **2015**, *66*, pp 269–278.
34. Y. Li, C. Zhao, Y. Qian, H. Wang, and H. Jin. High-Speed and Dense Three-Dimensional Surface Acquisition Using Defocused Binary Patterns for Spatially Isolated Objects. *Opt. Express* **2010**, *18*, pp 21628–21635.
35. Z. Li, K. Zhong, Y. Li, X. Zhou, and Y. Shi. Multiview Phase Shifting: A Full-Resolution and High-Speed 3D Measurement Framework for Arbitrary Shape Dynamic Objects. *Opt. Lett.* **2013**, *38*, pp 1389–1391.
36. K. Zhong, Z. Li, Y. Shi, C. Wang, and Y. Lei. Fast Phase Measurement Profilometry for Arbitrary Shape Objects Without Phase Unwrapping. *Opt. Lasers Eng.* **2013**, *51*, pp 1213–1222.
37. W. Lohry, V. Chen, and S. Zhang. Absolute Three-Dimensional Shape Measurement Using Coded Fringe Patterns Without Phase Unwrapping or Projector Calibration. *Opt. Express* **2014**, *22*, pp 1287–1301.
38. W. Lohry and S. Zhang. High-Speed Absolute Three-Dimensional Shape Measurement Using Three Binary Dithered Patterns. *Opt. Express* **2014**, *22*, pp 26752–26762.
39. Z. Zhang. A Flexible New Technique for Camera Calibration. *IEEE Trans. Pattern Anal. Mach. Intell.* **2000**, *22*, pp 1330–1334.
40. S. Zhang and P. S. Huang. Novel Method for Structured Light System Calibration. *Opt. Eng.* **2006**, *45*, p 083601.
41. R. Legarda-Sa, T. Bothe, W. P. Ju, et al. Accurate Procedure for the Calibration of a Structured Light System. *Opt. Eng.* **2004**, *43*, pp 464–471.
42. S. Zhang and S.-T. Yau. High-Resolution, Real-Time 3D Absolute Coordinate Measurement Based on a Phase-Shifting Method. *Opt. Express* **2006**, *14*, pp 2644–2649.
43. S. Zhang, D. Royer, and S.-T. Yau. GPU-Assisted High-Resolution, Real-Time 3-D Shape Measurement. *Opt. Express* **2006**, *14*, pp 9120–9129.
44. B. Li, N. Karpinsky, and S. Zhang. Novel Calibration Method for Structured Light System with an Out-of-Focus Projector. *Appl. Opt.* **2014**, *56*, pp 3415–3426.
45. B. Li and S. Zhang. Structured Light System Calibration Method with Optimal Fringe Angle. *Appl. Opt.* **2014**, *53*, 7942–7950.

46. S. Rusinkiewicz, O. Hall-Holt, and M. Levoy. Real-Time 3D Model Acquisition. *ACM Trans. Graph.* **2002**, *21*, pp 438–446.
47. O. Hall-Holt and S. Rusinkiewicz. Stripe Boundary Codes for Real-Time Structured-Light Range Scanning of Moving Objects, in *Proc. 8th IEEE International Conference on Computer Vision (ICCV 2001)*; Vol. 2, IEEE, 2001; pp 359–366.
48. Z. J. Geng. Rainbow Three-Dimensional Camera: New Concept of High-Speed Three-Dimensional Vision Systems. *Opt. Eng.* **1996**, *35*, pp 376–383.
49. K. G. Harding, M. P. Coletta, and C. H. Van Dommelen. Color Encoded Moiré Contouring, in *1988 Robotics Conference*; International Society for Optics and Photonics, 1989; pp 169–178.
50. P. S. Huang, Q. Hu, F. Jin, and F.-P. Chiang. Color-Encoded Digital Fringe Projection Technique for High-Speed Three-Dimensional Surface Contouring. *Opt. Eng.* **1999**, *38*, pp 1065–1071.
51. J. Pan, P. S. Huang, and F.-P. Chiang. Color Phase-Shifting Technique for Three-Dimensional Shape Measurement. *Opt. Eng.* **2006**, *45*, p 013602.
52. W.-H. Su. Color-Encoded Fringe Projection for 3D Shape Measurements. *Opt. Express* **2007**, *15*, pp 13167–13181.
53. S. Zhang and S.-T. Yau. High-Speed Three-Dimensional Shape Measurement System Using a Modified Two-Plus-One Phase-Shifting Algorithm. *Opt. Eng.* **2007**, *46*, p 113603.
54. L. J. Hornbeck. Digital Light Processing for High-Brightness High-Resolution Applications, in *Electronic Imaging '97*; International Society for Optics and Photonics, 1997; pp 27–40.
55. Y. Gong and S. Zhang. Ultrafast 3-D Shape Measurement with an Off-the-Shelf DLP Projector. *Opt. Express* **2010**, *18*, pp 19743–19754.
56. N. Karpinsky, M. Hoke, V. Chen, and S. Zhang. High-Resolution, Real-Time Three-Dimensional Shape Measurement on Graphics Processing Unit. *Opt. Eng.* **2014**, *53*, p 024105.
57. R. Raskar, G. Welch, M. Cutts, A. Lake, L. Stesin, and H. Fuchs. The Office of the Future: A Unified Approach to Image-Based Modeling and Spatially Immersive Displays, in *Proc. 25th Annual Conference on Computer Graphics and Interactive Techniques*; ACM, 1998; pp 179–188.
58. A. Jones, M. Lang, G. Fyffe, X. Yu, J. Busch, I. McDowall, M. Bolas, and P. Debevec. Achieving Eye Contact in a One-to-Many 3D Video Teleconferencing System. *ACM Trans. Graph.* **2009**, *28*, p 64.
59. S. Zhang and P. S. Huang. High-Resolution, Real-Time Three-Dimensional Shape Measurement. *Opt. Eng.* **2006**, *45*, p 123601.
60. N. Karpinskys. Portal-s: high-resolution, real-time 3D video telepresence. Ph.D. thesis, Iowa State University, 2013.
61. C. Guan, L. G. Hassebrook, and D. L. Lau. Real-Time 3D Data Acquisition for Augmented-Reality Man and Machine Interfacing, in *AeroSense 2003*; International Society for Optics and Photonics, 2003; pp 40–47.
62. C. J. Casey, D. L. Lau, and L. G. Hassebrook. *Structured Light Illumination Methods for Continuous Motion Hand and Face–Computer Interaction*. InTech, 2008.
63. G. Sandbach, S. Zafeiriou, M. Pantic, and L. Yin. Static and Dynamic 3D Facial Expression Recognition: A Comprehensive Survey. *Image Vis. Comput.* **2012**, *30*, pp 683–697.
64. N. Villaroman, D. Rowe, and B. Swan. Teaching Natural User Interaction Using OpenNI and the Microsoft Kinect Sensor, in *Proc. 2011 Conference on Information Technology Education*; ACM, 2011; pp 227–232.
65. M. Zollhöfer, M. Martinek, G. Greiner, M. Stamminger, and J. Süßmuth. Automatic Reconstruction of Personalized Avatars from 3D Face Scans. *Comput. Anim. Virtual Worlds* **2011**, *22*, pp 195–202.
66. Radiohead. Radiohead: House of Cards. Available at https://www.youtube.com/watch?v=8nTFjVm9sTQ, 2008.
67. G. Bhatia, M. W. Vannier, K. E. Smith, P. K. Commean, J. Riolo, and V. L. Young. Quantification of Facial Surface Change Using a Structured Light Scanner. *Plast. Reconstr. Surg.* **1994**, *94*, pp 768–774.
68. R. P. Mehta, S. Zhang, and T. A. Hadlock. Novel 3-D Video for Quantification of Facial Movement. *Otolaryngol. Head Neck Surg.* **2008**, *138*, pp 468–472.
69. O. V. Olesen, R. R. Paulsen, L. Hojgaard, B. Roed, and R. Larsen. Motion Tracking for Medical Imaging: A Nonvisible Structured Light Tracking Approach. *IEEE Trans. Med. Imaging* **2012**, *31*, pp 79–87.
70. J. I. Laughner, S. Zhang, H. Li, C. C. Shao, and I. R. Efimov. Mapping Cardiac Surface Mechanics with Structured Light Imaging. *Am. J. Physiol. Heart Circ. Physiol.* **2012**, *303*, pp H712–H720.
71. A. K. Iyengar, H. Sugimoto, D. B. Smith, and M. S. Sacks. Dynamic *In Vitro* Quantification of Bioprosthetic Heart Valve Leaflet Motion Using Structured Light Projection. *Ann. Biomed. Eng.* **2001**, *29*, pp 963–973.
72. H. Haneishi, T. Ogura, and Y. Miyake. Profilometry of a Gastrointestinal Surface by an Endoscope with Laser Beam Projection. *Opt. Lett.* **1994**, *19*, pp 601–603.
73. D. Stoyanov, A. Darzi, and G. Z. Yang. Dense 3D Depth Recovery for Soft Tissue Deformation During Robotically Assisted Laparoscopic Surgery. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2004*; Springer, 2004; pp 41–48.
74. W. W. Lau, N. A. Ramey, J. J. Corso, N. V. Thakor, and G. D. Hager. Stereo-Based Endoscopic Tracking of Cardiac Surface Deformation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2004*; Springer, 2004; pp 494–501.
75. M. Hu, G. Penney, P. Edwards, M. Figl, and D. J. Hawkes. 3D Reconstruction of Internal Organ Surfaces for Minimal Invasive Surgery. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2007*; Springer, 2007; pp 68–77.
76. L. Maier-Hein, P. Mountney, A. Bartoli, H. Elhawary, D. Elson, A. Groch, A. Kolb, M. Rodrigues, J. Sorger, S. Speidel, et al. Optical Techniques for 3D Surface Reconstruction in Computer-Assisted Laparoscopic Surgery. *Med. Image Anal.* **2013**, *17*, pp 974–996.
77. M. Boebel, A. Frank, K.-P. Hipp, R. Heimberger, F. Bitrolf, F. Haehnle, C.-S. Wagner, S. Prestel, and G. F. Buess. Stereo-endoscope with camera connection. U.S. Patent 8,608,645, 2013.
78. Y. Kazakevich and D. D. Sjostrom. 360 degree panning stereo endoscope. U.S. Patent Application 13/860,328, 2013.
79. J. Zobel. Stereo endoscope system. U.S. Patent 20150168710, 2015.
80. M. Schubert and A. Müller. Evaluation of Endoscopic Images for 3-Dimensional Measurement of Hollow Biological Organs. *Biomed. Tech.* **1998**, *43*, pp 32–33.
81. K. Armbruster and M. Scheffler. Messendes 3D-endoskop. *Horizonte* **1998**, *12*, pp 15–16.
82. B. Kemper, A. Merker, S. Lai, and G. von Bally. A Novel Electronic-Speckle-Pattern Interferometer (ESPI) for

Dynamic Holographic Endoscopy. In *Optics and Lasers in Biomedicine and Culture*, Vol. V; Springer, 1999; pp 29–32.

83. C. A. Bendall, G. Song, L. Tao, K. G. Harding, and T. Karpen. Fringe projection system and method for a probe suitable for phase-shift analysis. U.S. Patent 7,821,649, 2010.
84. C. Schmalz, F. Forster, A. Schick, and E. Angelopoulou. An Endoscopic 3D Scanner Based on Structured Light. *Med. Image Anal.* **2012**, *16*, pp 1063–1072.
85. H. Fuchs, M. A. Livingston, R. Raskar, D. Colucci, K. Keller, A. State, J. R. Crawford, P. Rademacher, S. H. Drake, and A. A. Meyer. Augmented Reality Visualization for Laparoscopic Surgery, in *Proc. 1st International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*; 1998; pp 934–943.
86. X. Maurice, C. Albitar, C. Doignon, and M. de Mathelin. A Structured Light-Based Laparoscope with Real-Time Organs' Surface Reconstruction for Minimally Invasive Surgery, in *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*; San Diego, CA, 2012; pp 5769–5772.
87. A. Reiter, A. Sigaras, D. Fowler, and P. K. Allen. Surgical Structured Light for 3D Minimally Invasive Surgical Imaging, in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014)*; IEEE, 2014; pp 1282–1287.
88. N. T. Clancy, D. Stoyanov, L. Maier-Hein, A. Groch, G.-Z. Yang, and D. S. Elson. Spectrally Encoded Fiber-Based Structured Lighting Probe for Intraoperative 3D Imaging. *Biomed. Opt. Express* **2011**, *2*, pp 3119–3128.
89. B. Li, P. Ou, and S. Zhang. Superfast 3D Optical Sensing with Fiber Interference. SPIE Newsroom, 2014.
90. A. K. Jain, P. Flynn, and A. A. Ross. *Handbook of Biometrics*. Springer, 2007.
91. P. J. Phillips, P. J. Flynn, T. Scruggs, K. W. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek. Overview of the Face Recognition Grand Challenge, in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005)*; Vol. 1, IEEE, 2005; pp 947–954.
92. A. Lagorio, M. Tistarelli, M. Cadoni, C. Fookes, and S. Sridharan. Liveness Detection Based on 3D Face Shape Analysis, in *2013 International Workshop on Biometrics and Forensics (IWBF)*; 2013; pp 1–4.
93. K. W. Bowyer, K. Chang, and P. Flynn. A Survey of Approaches and Challenges in 3D and Multi-Modal 3D + 2D Face Recognition. *Comput. Vis. Image Understand.* **2006**, *101*, pp 1–15.
94. H. Drira, B. Ben Amor, A. Srivastava, M. Daoudi, and R. Slama. 3D Face Recognition Under Expressions, Occlusions, and Pose Variations. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, pp 2270–2283.
95. K. W. Bowyer, K. Hollingsworth, and P. J. Flynn. Image Understanding for Iris Biometrics: A Survey. *Comput. Vis. Image Understand.* **2008**, *110*, pp 281–307.
96. J. Connell, N. Ratha, J. Gentile, and R. Bolle. Fake Iris Detection Using Structured Light, in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*; 2013; pp 8692–8696.
97. M. De Marsico, C. Galdi, M. Nappi, and D. Riccio. FIRME: Face and Iris Recognition for Mobile Engagement. *Image Vis. Comput.* **2014**, *32*, pp 1161–1172.
98. E. Bagci. Reverse Engineering Applications for Recovery of Broken or Worn Parts and Re-manufacturing: Three Case Studies. *Adv. Eng. Software* **2009**, *40*, pp 407–418.
99. V. Carbone, M. Carocci, E. Savio, G. Sansoni, and L. De Chiffre. Combination of a Vision System and a Coordinate Measuring Machine for the Reverse Engineering of Freeform Surfaces. *Int. J. Adv. Manuf. Technol.* **2001**, *17*, pp 263–271.
100. L. Li, N. Schemenauer, X. Peng, Y. Zeng, and P. Gu. A Reverse Engineering System for Rapid Manufacturing of Complex Objects. *Robot. Comput. Integr. Manuf.* **2002**, *18*, pp 53–67.
101. J. Gao, X. Chen, D. Zheng, O. Yilmaz, and N. Gindy. Adaptive Restoration of Complex Geometry Parts Through Reverse Engineering Application. *Adv. Eng. Software* **2006**, *37*, pp 592–600.
102. S. C. Park and M. Chang. Reverse Engineering with a Structured Light System. *Comput. Ind. Eng.* **2009**, *57*, pp 1377–1384.
103. J. A. Jalkio, R. C. Kim, and S. K. Case. Three Dimensional Inspection Using Multistripe Structured Light. *Opt. Eng.* **1985**, *24*, pp 246966–246966.
104. G. Zhang and Z. Wei. A Novel Calibration Approach to Structured Light 3D Vision Inspection. *Opt. Laser Technol.* **2002**, *34*, pp 373–380.
105. J. Gühring. Dense 3D Surface Acquisition by Structured Light Using Off-the-Shelf Components, in *Proc. SPIE*; San Jose, CA, 2000; pp 220–231.
106. L.-S. Bieri and J. Jacot. Three-Dimensional Vision Using Structured Light Applied to Quality Control in Production Line, in *Proc. SPIE*; Strasbourg, France, 2004; pp 463–471.
107. Y. Li, Y. F. Li, Q. L. Wang, D. Xu, and M. Tan. Measurement and Defect Detection of the Weld Bead Based on Online Vision Inspection. *IEEE Trans. Instrum. Meas.* **2010**, *59*, pp 1841–1849.
108. J. Xu, N. Xi, C. Zhang, Q. Shi, and J. Gregory. Real-Time 3D Shape Inspection System of Automotive Parts Based on Structured Light Pattern. *Opt. Laser Technol.* **2011**, *43*, pp 1–8.
109. R. Usamentiaga, J. Molleda, and D. F. Garcia. Structured-Light Sensor Using Two Laser Stripes for 3D Reconstruction Without Vibrations. *Sensors* **2014**, *14*, pp 20041–20063.
110. F. Bruno, G. Bianco, M. Muzzupappa, S. Barone, and A. Razionale. Experimentation of Structured Light and Stereo Vision for Underwater 3D Reconstruction. *ISPRS J. Photogram. Remote Sensing* **2011**, *66*, pp 508–518.
111. D. Zhan, L. Yu, J. Xiao, and T. Chen. Multi-camera and Structured-Light Vision System (MSVS) for Dynamic High-Accuracy 3D Measurements of Railway Tunnels. *Sensors* **2015**, *15*, pp 8664–8684.
112. S. Qiu, C. Cao, B. Zhang, X. Shao, W. Xie, Y. Bai, and C. Li. Feasibility Study of Remote Sensing Using Structured Light for 3D Damage Assessments After Natural Disasters. *Proc. SPIE* **2014**, *9263*, pp 92632R-1–92632R-10.
113. Y. Liu and G. Nejat. Multirobot Cooperative Learning for Semiautonomous Control in Urban Search and Rescue Applications. *J. Field Robot.* **2015**, doi:10.1002/rob.21597.

Tyler Bell
Beiwen Li
Song Zhang

Purdue University, West Lafayette, IN, USA